

O. Nurgeldiyew, G. Şamyradowa

***C++ DİLİNDE
PROGRAMMIRLEMEK***

Aşgabat - 2014

GIRIŞ

Kompýuter tehnologiýasy – iň ýaş ugurlaryň biridir. Kompýuter tehnologiýasynyň ösüş taryhy beýlekilere garaňda kän bir uly döwri alýan däl – 40-50 ýyl. Kompýuter tehnologiýasy, tehnika diýen adalgalary bolsa ondan hem ýaşdyr. Bilişimiz ýaly, ilki başda elektron-hasaplaýyş maşyn, hasaplaýyş tehnika diýen adalgalary ulanylardy. 20 ýyldan bäri bolsa EHM diýlen adalga ýuwaş-ýuwaşdan ýitip kompýuter adalga öz ornuny berdi, tehnika bolsa öňküler ýaly hasaplaýyş dälde kompýuter tehnika diýlip atlandyrylýar.

Kompýuter tehnologiýasy ýaş bolmak bilen, dünýäde öňdebaryjy ugurlaryň biri bolup durýar. Häzirki wagtda habarlar-aragatnaşyk tehnologiýalaryň ýokary depginde ösýändigini barada aýdylýar. Öýjükli telefon aragatnaşygyň, maglumat tehnologiýalaryň ösmegi muňa subut bolup durýar. Şol tehnologiýalaryň düzümine çuňlaýyn seredilen mahalynda olaryň kompýuter ugruna esaslanýandygyna göz ýetirmek bolýar.

Öz gezeginde, kompýuterler öz düzüminde tehnikaň başga ugurlarynyň soňky derejelerini jemländir. Bu bolsa onuň bilen işlemegi diňe ýeňillemän, eýsem amatly edip goýýar.

Türkmenistan dünýä ösüşiniň gapdalynda durman, kompýuter tehnologiýalaryň soňky gazananlaryny ulanmaklyga ymtylýar. Ýurdumyzda öňdebaryjy tehnologiýalary öwrenmeklik boýunça uly işler alnyp barylýar. Şol işlerde Hormatly Prezidentimiziň ýardam bermegi olaryň tiz depginde amala aşmagyny üpjün edýär. Ýurdumyzyň Baştutany öz gymmatly wagtyny tygşytlanman dünýäniň ösüşindäki ymtylyşlara üns berýär we olaryň has netijelilerini döwletimizde gerekli ugurlarda ornaşdyrylmagyna ýardam berýär.

Täze galkynyşlar zamanasynda ýurdumyzyň islendik pudagyň önünde täze meseleler goýuldy. Şol meseleleri üstünlikli çözmek üçin diňe bir tehnologiýalar ýeterlikli däl. Şol tehnologiýalary ulanyp biljek ýokary derejeli hünärmenler zerur.

C - programma düzmekligiň umumy dilidir. Bu dil UNIX sistemasy bilen berk baglanyşyklydyr, ýagny bu sistemada döredilendir, öz gezeginde UNIX sistemasynda işleýän programmalaryň aglabasy, şeýle hem UNIX sistemasynyň özüniň soňky wersiýalary C dilinde ýazylandyr. Ýöne diliň özi anyk bir maşyn bilen ýa-da operasion sistema bilen aýrylmaz baglanyşykly däl. Bu dil "sistemany programmirlemekligiň dili" diýip atlandyrylýar, sebäbi C dilinde kompilýatorlary we operasion sistemalary yazmak has amatly, şeýle hem bu dilde beýleki uly programma önümlerini döretmek üçin hem juda giň mümkinçilikler bar.

C dili döredilende köp wajyp ideýalar Martin Riçards tarapyndan döredilen BCPL dilinden alnypdyr. Ol täsir 1970-nji ýylda Ken Tompson tarapyndan PDP-

7-de döredilen, UNIX-iň ilkinji sistemasy üçin niýetlenen B diliniň üsti bilen geçipdir.

BCPL we B "tipsiz" dillerdir. Olardan tapawutlylykda C dili maglumatlaryň dürli tipleri bilen üpjün edýär. Baza tipine harplar(char), şeýle hem dürli ölçegli bitin we hakyky sanlar girýär. Bulardan başgada görkezijilerden(pointer), massiwlerden(array), gurluşlardan(structure) we birleşmelerden(?) programmistiň özüne tipleriň giden toplumyny döretmäge mümkinçik berilýär. Aňlatma operatorlardan we operandlardan düzülýär. Islendik aňlatma, eýe bolmak we funksiýany çagyrmak aňlatmalaryndan başlap hemmisi instruksiýa(görkezme) bolup biler. Görkezijiler maşyna bagly bolmadyk adres arifmetikasy bilen üpjün edýär.

C-de gowy gurluşly(strukturaly) programmalarda ulanylýan, esasy dolandyryjy konstruksiýalar bar: düzme instruksiýa ({...}), şert boýunça şahalanmak(if-else), birnäçe ýoldan birini saýlap almak(switch), şerti başda barlanylýan gaýtalanmak operatory(while, for) we şerti soňynda barlanylýan gaýtalanmak operatory(do), şeýle hem gaýtalanmaklygyň arasyny üzyň serişde(break).

1. C++ DILI BILEN TANYŞLYK

Programma ýazmagyň düzgüni

Function-lara(kömekçi programma) ýüzlenilenden soňra, olar netije görnüşinde baza tiplerini, structuralary, birleşmeleri we görkezijileri berip bilerler. Islendik function özüne rekursiw ýüzlenip biler. Düzgün bolşy ýaly, function-ýň lokal üýtgeýän ululyklary funktion-a her gezek ýüzlenilende “awtomatiki”, ýagny gaýtadan döredilýär. Bir function-y beýleki function-ýň içinde kesgitlep bolmaýär, ýöne üýtgeýän ululyklaryň deklarasiýasyny blok-struktura görnüşinde kesgitlemäge mümkinçilik berilýär. C dilinde ýazylan programmanyň function-lary aýratyn faýlda saklanyp we aýratynlykda kompilirlenip biliner. Functionlardaky üýtgeýän ululyklar içki we daşky üýtgeýän ululyklara bölünýärler. Daşky üýtgeýän ululyklar bir faýlyň çäginde ýa-da bütün programmada ulanylyp biliner.

Preprosessirlemek etapynda programmanyň tekstiniň içine daşyndan makroslar, beýleki programmalar(ishodnik) gaýulýar we şertleýin kompilirlenýär.

C dili beýleki köp dillere görä "pes derejeli" dil hasaplanýar. Ýöne munuň özi onuň abraýyny peseltmeýär, ol kompýuterleriň köpüsiniň iş salyşýan obýektleri bolan, ýagny harplar(char), sanlar, we adresler bilen işleýär. Olar bilen real maşynda bar bolan arifmetiki we logiki operasiýalary ulanmak arkaly iş salyşmaklyga mümkinçilik berýär.

C dilinde setirler(string), köplükler, spisoklar, massiwler ýaly düzme obýektleriň üstünden geçirilýän göni operasiýalar ýok. Bu dilde structure-lara(gurluş) bir obýekt hökmünde seredilip olaryň tutuş kopiýasyny almak bolýan hem bolsa, bitin tipli massiwler ýa-da setir ululyklary bilen işlemäge mümkinçilik berýän operasiýalar ýok. Funksiýanyň lokal üýtgeýän ululyklary üçin ýer bölünip berlende ulanylýan static üýtgeýän ululyklar we stack mehanizminden başga kompýuteriň ýadyny paýlamaga mümkinçilik berýän serişde ýok. Şeýle hem dürli maglumatlary ýerleşdirmek üçin ulanylýan "heap-ýady"(памяти-кучи) we "hapany ýygnaýjy"("сборщик мусора") ýok. C diliniň özünde giriş-çykyş serişdesi, ýagny READ(okamak) we WRITE(ýazmak) instruksiýalary ýa-da faýllara ýüzlenmekligiň başga bir usuly ýok. Bularyň hemmesi – ýokary derejeli mehanizm bolup, C dilinde funksiyalary açyk çagyrmak arkaly amala aşyrylmalydyr. C ulgamynyň(diliniň) köp görnüşinde gerek bolaýjak şeýle function-laryň toplumu bar.

Ýokardaky aýdylanlary dowam etmek bilen belläp geçmeli zat, C dili hasaplamany gidişini diňe yzygiderli dolandyrmaga mümkinçilik berýän serişde bilen üpjün edýär: şerte görä şahalanmak mehanizmi, gaýtalanmak, düzme instruksiýa, kömekçi programma. Multiprogrammirlemek, parallel prosesler, sinhron we programmany gurnamak serişdelerini bolsa özünde saklamalýar.

Ýokardaky agzalyp geçilen serişdeleriň bolmazlygy uly kemçilik ýaly bolup görünýär(“iki setiri deňeşdirmek üçin function-a ýüzlenmeli bolýar”). Ýöne diliň gysgalygynyň(ykjamlygynyň, kompaktlygy) hakyky peýdasy iş ýüzünde ýüze çykýar. Umuman aýdylanda C dili beýleki diller bilen deňeşdureninde göwrümi boýunça az, şeýlelikde onuň düşündirilişi hem gysga, şonuň üçin ony beýleki dillere garanynda az wagtda öwrenmek bolar. Özünem programmist şol az wagtda C diliniň hemme mümkinçiliklerini öwrenjekligine, düşünjekligine we praktikada ulanyp biljekligine bil baplap biler.

Köp ýyllaryň dowamynda “C programmirlemek dili” atly kitabyň, birinji neşiri bu dil baradaky ýeke-täk standart(kesgitleme, kitap) bolup hyzmat edip geldi. 1983-nji ýylda amerikanyň milli standartlar instituty(ANSI) tarapyndan C diliniň döwrebap standartyny(kesgitlemesini) taýýarlamak boýunça komitet döredilýär. Ol komitet tarapyndan 1988-nji ýylda C diliniň standarty(“ANSI-C”) taýýarlanylýar. Ol standartyň köp düzgüni häzirki zaman kompilýatorlarynda ulanylýar.

Standart ilkinji çykan sorag-jogap gollanmasyna esaslanýar. Ol gollanma bilen deňeşdirilende dil az özgeripdir. Standartyň esasy maksatlarynyň biri, bar bolan programmalaryň köp böleginiň kabul edilen standarta gabat gelmegidir ýa-da ol programmalaryň standarta gabat gelmeýän ýerleri barada kompilýatoryň duýdurýjy habarlary bermegidir.

Programmistleriň köpüsi üçin function-y beýan etmegiň we kesgitlemegiň täze sintaksisi kabul edilen standartdaky iň wajyp täzelik boldy. Indi function bilen birlikde onuň argumentleri hem kesgitlenip biliner. Şeýlelikde function-y kesgitlemekligiň sintaksisi hem üýtgedi. Argumentleriň kesgitlenmegi bilen ýüze çykýan goşmaça maglumat, tipleri gabat gelmeýän argumentler bilen baglanyşykly näsazlyklary tapmakda kompilýatoryň işini has ýeňilleşdirýär; munuň özi biziň pikirimizçe dile edilen örän peýdaly goşundydyr.

Ýene-de uly bolmadyk täzelikleriň bir toparyny belläp geçmek artykmaçlyk etmez. Dildäki esli wagtda bari giňden ulanylyp gelinýän structure we enaum(?) tipi düzgünleşdirildi. Indi süýşýän nokat bilen hasaplamak birlik takyklykda hem ulanylyp biliner. Umuman arifmetikanyň häsiýeti, esasan hem alamatsyz tipler üçin aýdyňlaşdyryldy. Preprocessor kämilleşdirildi. Programmistleriň köpüsi bu üýtgeşmeleri örän gowşak ulanýarlar.

Standartyň ikinji bir aýdyp geçmeli esasy tarapy, operasion sistemanyň mümkinçiliklerini ulanmaga ygtyýar berýän(mysal üçin faýllary okamaga we ýazmak), formatly giriş-çykyş, ýady dinamiki ulanmaklyk, string(literlerden düzülen zynjyrjyk) bilen işlemeklik üçin gerek bolan, C kompilýator bilen goýulýan bibliotekalaryň kesgitlenmegidir. Standart baş faýllaryň toplumynyň bolmagy, maglumatlaryň tipleriniň we functiony beýan etmekligiň birmeňzeş usuly bilen üpjün edýär. Bu bibliotekalary ulanýan programmalar şol operasion sistemany ulananda, islendik başga maşynlarda hen ýerine ýetjekdigine güwä geçilýär. Bibliotekany düzýän köp programmalar UNIX sistemasynyň "standart giriş-çykyş biblioteka"-synyň nusgasynda we meňzeşlikde döredilendir. Bu biblioteka kitabyň birinji neşirinde ýazylyp görkezilýär we beýleki sistemalarda hem giňden ulanylýar. Bu ýerde hem programmistler düýpli tapawudy duýmazlar.

C diliniň maglumatlarynyň tipi we dolandyryjy düzümi(strukturasy) bar bolan maşynlaryň köpüsiniň komandalary bilen gollanylýar, programmanyň özbaşdak göyberilmeginiň we hasalama geçirilmeginiň administratiw sistemasy bolsa örän seýrek duşýar. Bibliotekadaky functionlara ýüzlenmekligi programmist ýazýar(kompilýator däl), şonuň üçin hem isleg bolanda olary beýlekilere çalyşmak bolar. Operasion sistemadaky käbir gizlin detallar bilen bagly aýratynlyklary ulanmaýan C dilinde ýazylan programmalaryň hemmesi diýen ýaly, islendik maşynda ýerine ýetýär.

C dili belli bir, konkret maşynyň arhitekturasy göz önünde tutulyp ýazylan däl, ýöne ol köp maşynlaryň apparat mümkinçiliklerine gabat gelýär. Belli bir kadalara eýermek bilen, islendik maşynda ýerine ýetýän C programmany ýazmak bolar. Standart, programmanyň ýerine ýetjek maşynynyň karakteristikasy görkezýän hemişelikleriň toplumyny ulanmak bilen, islendik maşynda ol programmanyň ýerine ýetmekligini gazanmaklyga mümkinçilik döredýär.

C "berk tipleşdirilen" dil bolmasa hem, ol öz ösüş döwründe tipleriň üstünden barlagy güýçlendirdi. C diliniň ilkinji wersiýalarynda, bitin sanlar bilen görkezijileriň arasynda barlagsyz çalyşmaklyga rugsat edilýärdi, munu özi köp düşnüksizlige sebäp bolýardy, şonuň üçin hem C diliniň soňky wersiýalarynda beýle zat gadagan edildi. Standarta baglylykda, täze kompilýatorlar ulanylýan maglumatyň tipini we tipleriň özgerdilmesini aýk görkezmekligi berk talap edýär. Functiony beýan etmekligiň täze görnüsi - bu tarapa ädilen ýene bir ädimdir. Indi kompilýator tiplerdäki ýalňyşlyklaryň köp bölegi barada duýduryş berýär we ylalaşmaýan tipleri awtomatiki özgertermekligi amala aşyрмаýar. Programmistler öz etjek işini bilýändir - diýen C-diliniň esasy pelsepesi öz güýjünde galyp, diňe meseläniň çözülmegi üçin has anyklaşdyrmaklygy talap edýär.

C dili hem beýleki diller ýaly kemçiliklerden halas däl. Käbir operatorlaryň ululyk derejesi umumy kabul edilen däl, käbir sintaksis gurluşlar has hem gowy edilip gurnalyp bilinerdi. Şeýlede bolsa C dili, köp meseleleri çözmek üçin amatly, adatdan daşary effektiv we düşnükli dildir.

Kitabyň şeýle düzümi bar. 1-nji bap C diliniň esasy serişdelerine syn. Ol okyjynyň mümkin boldugyndan çalt programma ýazmaklygyna höwes döreder, ýagny biziň ynanyşmyza görä täze dili öwrenmekligiň ýeke täk usuly ol dilde programma ýazmakdyr. Kitabyň bu bölegine oňat düşünmek üçin, programmirlemekligiň esasy elementlerinden habarly bolunsa has oňat bolar. Bu ýerde kompýuter näme, kompilyasiýa näme ýa-da $n=n+1$ aňlatmanyň nämäni aňlatýanlygy barada hiç-hili düşündiriş berilmeýär. Mümkin bolan ýerde programmirlemekligiň peýdaly usullaryny görkezmäge çalyşan hem bolsak, bu kitabyň maglumatlaryň düzümi boýunça-da, algoritmler boýunça-da sorag-jogap gollanmasy bolmaklygy göz-önünde tutulmady; bu ýerde esasan nämä ünsi çekilýär diýilende bolsa, diliň özüne ünsi çekilýär - diýilse has dogry bolar.

2-6-njy baplarynda bolsa diliň dürli serişdelerine 1-nji bapdaka garanda has çintgiläp we has formal(umumy) seredilip geçilýär; özünem bu ýerde hem edil öňki ýaly izolirlenen böleklere dälde, gutarnykly programmalar bolan mysallara ünsi çekilýär. 2-nji bapda maglumatlaryň baza tipi, operatorlar we aňlatmalar bilen tanyşdyrylýar. 3-nji bapda hasaplamaklygyň yzygiderligini dolandyrmaklygyň

serişdelerine seredilip geçilýär: if-else, switch, while, for we ş.m. 4-nji bapda function we programmanyň düzümine(daşky üýtgeýän ululyklar, üýtgeýän ululyklaryň görüniş düzgüni, programmany birnäçe faýllara bölmeklige e ş.m.), şeýle hem preprocessor barada gürrüň berilýär. 5-nji bapda görkezijiler we adres arifmetikasy meselesine seredilip geçilýär. 6-njy bap bolsa structura we birleşmä bagşylanandyr.

7-nji bapda, operasion sistema bilen umumy interfeýsi üpjün edýän standart biblioteka barada gürrüň berilýär. Bu biblioteka ANSI standartyna girizilipdir, başgaça aýdanymyzda bu biblioteka C dili bar bolan maşynlaryň ählisiniň düzümine girýär, şoňa göräde operasion sistemanyň giriş-çykyşyny we beýleki mümkinçiliklerini ulanýan programmany hiç-hili üýtgetmezden bir maşyndan beýleki maşyna geçirmek bolar.

8-nji bapda C dilinde ýazylan programmalar bilen UNIX operasion sistemasynyň arasyndaky interfeýsi, esasan hem giriş-çykyş, faýl sistemasy we ýadyň paýlanyşy baradaky mesele gozgalýar. Bu babyň birnäçe bölümi UNIX sistemasynyň aýratynlyklaryny bilen baglanyşykly bolsa-da, beýleki sistemalarda işleýän programmistler hem bu ýerde standart bibliotekanyň döredilişi barada, ýazylan programmanyň dürli maşynlarda ýerine ýetmegi üçin näme etmelidigi barada köp peýdaly maglumatlary taparlar.

Kitabyň soňunda berilýän A goşmaça sorag-jogap bölümçesi hökmünde seretmek bolar. C diliniň sintaksisiniň we semantikasynyň berk kesgitlemesi ANSI standartynyň ýörüte dokümentinde saklaýar. Bu dokument esasan hem kompilyatory döredijiler üçin has hem zerurdyr. Biziň sorag-jogap bölümçämiz standartyň ulanýan kanunalaýyk stillerine köp üns bermän dile gysgajyk kesgitleme berýär. B goşmaça - standart bibliotekalar barada maglumatlary özünde saklaýar we ol kompilyator döredijiler üçin däl-de esasan ulanyjylara(programmaisler) niýetlenendir. C goşmaça bolsa diliň birinji wersiýasyndakydan tapawudynyň gysgaça beýanyňy özünde saklaýar. Ýöne düşnüksiz ýagdaýlarda dil boýunça gutarnykly netije çykarýan emin, diliň standarty we siziň ulanýan kompilyatoryňyz bolup galýar.

C++ diliniň mümkinçiliklerine syn

C dili bilen gysgaça tanyşmaktan başlalyň. Biziň maksadymyz - ownuk-uşak zatlara, formal düzgünlere üns bermezden real programmalaryň üsti bilen diliň elementlerini açyp görkezmekden ybarat. Şonuň üçin hem biz dolylyga hem-de takyklyga ymtylmaýarys(ýöne mysallaryň korrektligi barada alada etdik). Eýle çemeleşmek bilen biz siziň peýdaly programmalary ýazjak wagtyňyzy ýakynlaşdyrmak isledik. Şonuň üçin siziň ünsiňyzy esasy zatlara: üýtgeýän ululyklara, hemişeliklere, arifmetika, hasaplamaklygyň yzygiderligini dolandyrmaklyga, funksiýalara we ýonekeý giriş-çykyşa jemlemek zerurdyr. Bu bapda uly programmalary ýazmak üçin gerek bolan serişdelere seretmeris. Olar görkezijiler, structure-ler, operatorlaryň köp bölegi, birnäçe dolandyryjy görkezmeler(instruksiýa) we standart bibliotekalar.

Beýle çemeleşmekligiň hem öz ýetmezçiligi bar. Olaryň biri hem biz bu ýerde diliň häsiýetini gutarnykly beýan edilmeýäris bu bolsa öz gezeginde käbir ýagdaýlara nädogry düşünmeklige getirip biler. Berlen materiallaryň çäklidigi üçin, biz seredip geçen mysallarymyzda diliň doly kuwwadyny ulanyp bilmedik, ýogsam olar has gysga hem gelşikli ýazylyp bilinerdi. Biz başardygymyzdan bu ýetmezçiligi duýulmaz ýaly edip bermeklige çalyşdyk, ýöne bu bapda ýazjak programmalarymyzda şeýle ýetmezçiligiň bardygyny aýdyp geçmek gerek. Beýleki bir ýetmezçilik, geljekgi baplarda käbir zatlary gaýtalap geçmeli bolarys. Ýöne gaýtalamakdan geljek peýda, geljekde oňa sarp eden wagtymyza ödär diýip umyt edýäris.

Islendik ýagdaýda hem tejribeli programmist bu babyň materiallaryndan öz düzyň programmalarynda ulanar ýaly peýdaly zatlary tapyp biler. Öwrenje programmistler bolsa, diňe bu ýerde ýazylyp görkezilen programmalary barlap görmek bilen çäklenmän, özüleri hem birnäçe kiçiräjik programmalary ýazyp görmek maslahat berilýär. Tejribeli programmistler hem, öwrenje programmistler hem bu babyň materiallaryna ullaňan sütünler hökmünde seretmelidirler, 2-nji bapdan başlap ol sütünlerde diliň elementlerini asylyp goýuljakdyr.

Programmirlemek dilini öwrenmekligiň ýeke täk usuly - munuň özi ol dilde programma ýazmakdan ybaratdyr. Islendik dil öwrenilende-de düzgün bolşy ýaly ilki bilen takmynan aşakdaky ýaly programma düzmeklikden başlaýarlar:

Salam ! sözünü çapa çykarýan programma ýazmaly.

Ine ilkinji kynçylyklar: haýsydyr bir tekst redaktorynda programmanyň tekstini taýýarlamaly, ony kompilirlemeli soňra bolsa ýerine ýetirmeli we netijesiniň niräk çykjaklygyny kesgitlemeli. Bu zatlary öwreneniňizden soň, galan zatlar bize aňsat ýaly bolup görner.

Aşakdaky programma “Salam!” sözünü çap edýär:

```
#include <stdio.h>
main()
{
    printf(“Salam!\n”);
}
```

Bu programmany işe göýbermek siziň ulanýan sistemaňyza bagly. UNIX operasion sistemasynda programmany “.c” simwol bilen gutarýan faýlda ýerleşdirýärler, mysal üçin hello.c. Soňra bu faýly

cc hello.c

komandanyň kömegi bilen kompilirleýärler. Egerde siz belgileriň hemmisini ýerbe-ýer goýan bolsaňyz, orfografik ýalňyşlyk goýbermän hemme zatlary dogry ýerine ýetiren bolsaňyz, onda kompilirlemeklik “sessiz-üýnsiz” geçer. Netije-de ýerine ýetirmeklige taýyn a.out atly faýly alarsyňyz. Indi siz ol faýly

a.out

komandasynyň kömegi bilen ýerine ýetirseňiz, programma

salam!

sözünü çap eder.

Beýleki sistemalarda programmany işe goýbermekligiň tertibi başgaça bolup biler. Ol barada degişli hünärmenler bilen maslahatlaşmaly.

Indi programmanyň özüne degişli käbir pursatlary aýdyňlaşdyrallyň. C dilinde ýazylan programma özüniň tutýan göwrümüne bagly bolmazdan function-lardan we üýtgeýän ululyklardan ybarat bolýar. Function-lar hasaplamak üçin gerek bolan görkezmeleri(komandalary, instruksiýalary) özünde saklaýan bolsa, üýtgeýän ululyklar hasaplama döwründe ulanylýan bahalary özünde saklaýarlar.

C dilindäki function-lar FORTRAN dilindäki kömekçi programmalara we function-lara, şeýle hem PASCAL dilindäki function-lara we Procedure-lere meňzeşdir. Ýokardaky görkezilen programma – munuň özi **main** atly functiondyr. Adatça siň öz ýazan function-laryňyza islendik at dakyp bilersiňiz, ýöne **main** – aýratyn at bolup, islendik programma şol function-ynyň ilkinji görkezmelerini ýerine ýetirmek bilen öz işine başlaýar.

Adatça **main** öz işini ýerine ýetirmek üçin beýleki function-laryň hyzmatyndan peýdalanýar; olaryň käbirlerini programmistiň özi döredýär, beýlekileri bolsa onuň ygtyýarynda bolan gordan(library, biblioteka) alynýar. Programmanyň ilkinji setiri:

```
#include <stdio.h>
```

kompilýatora standart giriş-çykyş goryny birleşdirmelidigi hakynda görkezme berýär. Bu setir köp C programmanyň asyl nusgasyny saklaýan faýlyň başynda gabat gelýär. Standart(adaty) gor 7-nji bapda we B şoşmaçada beýan edilýär.

Function-laryň arasyndaky maglumat geçirmekligiň bir usuly hem bir function beýlekä ýüzlenende argument diýilip atlandyrylýan birnäçe bahalaryň toplumyny geçirmegidir. Ol ululyklaryň aralary otur bilen bölünip ýaý şekilli skobkanyň içinde, function-yň adynyň yzynda ýerleşdirilýär. Biziň mysalymyzda **main** function görnüşinde kesgitlenýär, bu ýerde ýaý şekilli skobkanyň içiniň boşlugy () onuň hiç-hili argumente garaşmaýanlygyny aňladýar.

C dilinde ilkinji programma

```
#include <stdio.h>
main()
{
    printf("Hello, World\n");
}
```

Bu ýerde:

#include <stdio.h> setiri, standart bibliotekany programmamyza birleşdirýänligimizi aňladýar.

main() hiç-hili argumenti bolmadyk main atly function-y kesgitleýäris. Bu function-yn görkezmeleri(komandalary) figuraly ýaýlaryň içinde ýerleşdirilýär.

printf("Hello, World\n"); main function-y simwollaryň berlen yzygiderligini çapa çykarmak üçin birleşdirilen bibliotekanyň **printf** function-yna ýüzlenýär. \n täze-setire geçmekligi görkezýän litera(simwol).

Function-yň komandalary figuraly ýaýlaryň içinde ýerleşdirilýär. Bu ýerde main functiony diňe bir görkezmäni özünde saklaýar:

```
printf("Hello, World\n");
```

Function ady boýunça çagyrylýar we adynyň yzynda ýaý şekilli skobkanyň içinde argumentleriň sanawy görkezilýär. Şeýlelikde bu setirde **“Hello, World\n”** argument bilen **printf** functiony çagyrylýär. **printf** – biblioteka functiony bolup ol häzirki ýagdaýda goşa dyrnagyň içinde ýerleşdirilen simwollary çapa çykarýar.

“Hello, World\n” görnüşdäki goşadyrnagyň içinde ýerleşdirilen simwollaryň yzygiderligine *string*(setir) ýa-da *string hemişeligi* diýilýär. Belli bir wagtlap printf we beýleki functionynlaryň argumenti görnüşinde diňe stringleri ulanarys.

C-de \n literleriň birleşmesi stringiň içinde täze setir literasyny aňladýar, ol çapa çykarylanda indiki setiriň çep çäğine geçmekligi amala aşyrýär. Eger siz \n literany aýyrsaňyz(tejribede barlap bilersiňiz), onda çapa çykarmadan soň täze setire geçilmeýändigine göz ýetirersiňiz. Setir argumentinde täze setir(\n) literasynda açık görkezmeli. Mysal üçin siz aşakdaky ýaly ýazmak bilen:

```
printf(“Hello, World  
”);
```

indiki setire geçmekçi bolsaňyz onda kompilýator ýalňyşlygyň bardygy barada habar berer.

Täze setir literasy hiç-haçan awtomatiki goýulmaýar, şonuň üçin bir setiri printf functionyna birnäçe gezek ýüzlenmek bilen hem çapa çykarmak bolar: Biziň ilkinji programmamyzy aşakdaky ýaly hem ýazmak bolar:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    printf(“Hello, ”);
```

```
    printf(“World”);
```

```
    printf(“\n”);
```

```
}
```

Bu programmany ýerine ýetirsek öňki programmany ýerine ýetirenimizdäki setir çapa çykar.

\n diňe bir literany aňladýanlygyny belläliň. Esc yzygiderlik diýilip atlandyrylýn, \n ýaly ýapgyt çyzyk bilen başlanýan literalaryň aýratyn birleşmesi, göz öňüne getirmesi kyn ýa-da görünmeýän literalary bellemek üçin giňden ulanylýar. Beýleki literalar bilen birlikde ulanylýan \t, \b, \', \\ literalar degişlilikde tabulýasiýany, bir litera yza gaýtmak(backspace), goşa dyrnak, ýapgyt çyzygyň özüni aňladýar. Şeýle literalaryň doly sanawy 2.3 bölümde görkezilendir.

Ýumuş 1. Öz sistemaňyzda **“Hello, World”** setiri ekrana çykarýan programmany ýerine ýetiriň. Programmanyň käbir böleklerini aýyrmak bilen dürli tejribeleri geçirip görüň, ýalňyşlyklar barada berilýän habara üňs beriň.

Ýumuş 2. printf functionyň setir argumentine ýokarda görkezilen sanawda bolmadyk \c literany goýsak näme bolýandygyny barlap görüň(bu ýerde c ýokardaky sanawda bolmadyk litera).

Üýtgeýän ululyk we arifmetiki aňlatma

Indiki programma $^{\circ}\text{C}=(5/9)(^{\circ}\text{F}-32)$ formula boýunça hasaplamany geçirýär we aşakdaky temperaturanyň Ferengeýt we Selsiý boýunça gabat gelmek tablisasyny çapa çykarýar:

0	-17
20	-6
40	4
60	15
80	26
100	37
120	48
140	60
160	71
180	82
200	93
220	104
240	115
260	126
280	137
300	148

Öňki ýazan programmamyz ýaly bu programma hem ýeketäk main() functiondan ybarat. Bu programma “Hello, World” setiri çap edýän prigrammadan uzyn, ýöne kyn däl. Biz bu programmada kommentariýalary (düşündirişleri), üýtgeýän ululyklary beýan etmekligi, arifmetiki aňlatmalaryň ýazylyşyny, gaýtalanmaklygy we formatly çap etmekligi görkezeris:

```
#include <stdio.h>
/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý boýunça
tablisyny çap etmek */
main()
{
    int fahr, celsius;
    int    lower, upper, step;

    lower = 0;          /* temperatyralaryň tabl. aşaky çägi */
    upper = 300;        /* ýokarky çäk */
    step = 20;          /* ädim */

    fahr = lower;
    while (fahr <= upper) {
        celsius = 5 * (fahr-32) / 9;
        printf("%d\t%d\n", fahr, celsius);
        fahr = fahr + step;
    }
}
```

/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý boýunça tablisyny çap etmek */

bu iki setir *kommentariýa*(düşündiriş) bolup olar programmanyň ýerine ýetirýän işini gysgaça düşündirýär. /*, */ belgileriň arasynda ýerleşdirilen literalar kompilyator tarapyndan seredilmeyär we programmanyň düşnükli bolmagy üçin erkin ulanylyp biliner. Kommentariýa programmanyň islendik ýerinde ýerleşip, ol özünde boşluk literasyny, tabulýasiýa we täze setir literasyny saklap biler.

C dilinde islendik üýtgeýän ululyk ulanylmazyndan öň onuň haýsy tipe degişlidigi ýazylyp beýan edilmelidir; adaty hemme üýtgeýän ululyklar functionyň başynda ilkinji ýerine ýetýän instruksiýadan öň beýan edilýär. Deklarasiýada(beýan edilişde) üýtgeýän ululygyň hasiýeti aýan edilýär. Ol tipiň adyndan we üýtgeýän ululyklaryň toplumyndan(spisogyndan) ybaratdyr, mysal üçin:

Int fahr, celsius;

Int lower, upper, step;

Int tipi görkezilen üýtgeýän ululyklaryň bitin sanly ululyklardygyny aňladýar, ondan tapawudylykda **float** tipi ýüzyň nokatly üýtgeýän ululyklary görkezýär, ýagny drob bölegi bolup biljek sanlary görkezýär. **Int** hem-de **float** tipiň alyp biljek bahalarynyň diapazony ulanylyan maşyna bagly.

Int tipi 16-razryadly(ol -32768-den +32767-ä çenli diapazonda ýatýar), şeýle hem 32-razryadly bolup bilýär. **float** tipli sanlar adaty 32-razryadly sözler bilen ýazylýar, azyndan 6 sany manyly onluk sifri bolýar, 10^{-38} -den 10^{+38} -çenli diapazonda ýatýar.

C dilinde maglumatlar üçin **int** we **float** tiplerden başgaca birnäçe baza tipleri ulanmaklyga mümkinçilik berýär, olar:

char 1 - baýtly litera

short gysga bitin

long uzun bitin

double ikeldilen takyklykdaky ýüzyň nokatly

Görkezilen tipleriň obýektleriniň ölçegleri hem ulanylýan maşyna baglydyr. Baza tiplerden: *massiwleri*, *strukturalary* we *birleşmeleri* (объединение), netije görnüşinde ol tipleriň bahalaryny yzyna gaýtarýan baza tiplerinden bolan obýektlere we functionlara *görkezijileri* döretmek bolýar. Olar hakynda soňra gürrüň ederis.

Temperaturany özgertmek programmasyndaky hasaplamak aşakdaky eýe bolmak instruksiýalaryndan başlanýar:

lower = 0;

upper = 0;

step = 20;

fahr = lower;

olar atlary görkezilen üýtgeýän ululyklaryň başlangyç bahasyny kesgitleýär. Islendik instruksiýa nokatly otur bilen gutarýar.

Tablisanyň hemme setirleri şol bir usul bilen hasaplanýar, şonuň üçin her bir setir üçin hasaplamany gaýtalanmaklygy ulanylyp ýerine ýetireliň. while

gaýtalanmasynyň ýerine-ýetmeginiň zerur şerti:

```
while (fahr <=upper) {
```

```
...
```

```
}
```

Ol şeýle ýerine ýetýär. Skobkalaryň içindäki şert barlanylýar. Eger ol çyn(true) bolsa(fahr ululygyň bahasy upper ululygyň bahasyndan kiçi ýada deň bolsa), onda gaýtalanmaklygyň göwresi(figuraly skobkalaryň içinde ýazylan üç instruksiýa) ýerine ýetýär. Soňra ýene-de şert barlanylýar we ol çyn bolsa gaýtalanmaklygyň göwresi ýenede ýerine ýetýär. Haçanda şert ýalan(false, ýagny fahr upperden uly bolanda) gaýtalanmak tamamlanýar we hasaplamak gaýtalanmaklygyň yzyndaky instruksiýadan başlanýar. Biziň musalymyzda görnüşi ýaly gaýtalanmaklykdan soň hiç-hili instruksiýa ýok, şonuň üçin programma öz işini tamamlýar.

while instruksiýasynyň göwresi bir ýa-da temperaturany özgertmek programmasyndaky ýaly figuraly skobkanyň içinde ýazylan birnäçe instruksiýadan, ýa-da aşadaky ýalyskobkasyz ýazylan ýeke-täk instruksiýadan ybarat bolup biler:

```
while (i<j)
```

```
i=2*i;
```

Ol ýa-da beýleki ýagdaýda hem biz while gaýtalanmaklygynyň dolandyryýan instruksiýalaryny bir tabulýasiýa süýşürüp ýazarys, ol bolsa programmada 4 sany boşluga deňdir(пробел); şonuň üçin hem gaýtalanmaklygyň göwresini düzýän instruksiýalar aýyl-saýyl görünýärler. Boşluklar programmanyň logiki düzümini kesgitleýär. C kompilýator programmanyň daşky oformleniýasyna üns bermeýär, ýöne gerek ýerinde goýlan tabulýasiýalar we boşluklar programma seredenimizde onuň düşnükli bolmagyna ýardam edýär. Aňlatmalaryň logiki düzüminiň görünmegi üçin olary aýratyn setirde we operasiýa bellikleri boşluklar bilen gurşamagy maslahat berýäris. Skobkalaryň ýerleşşi esasy däl, bu barada dürli pikirler bar. Biz olaryň birnäçe görnüşleriniň ulanylyşyndan biriniň üstünde durup geçdik. Olaryň içinden siz öz halanyňyzy saýlap alyň we şonuň bilen işlemegi dowam etdiriň.

Hasaplamanyň köp bölegi gaýtalanmaklygyň göwresinde ýerine ýetirilýär. Temperatura

```
celsius = 5 * (fahr-32) / 9;
```

instruksiýanyň kömegi bilen Selsiý şkalasyna geçirilýär we celsius üýtgeýän ululygyna dakylýar. Biz bada-bat 9/5-e köpeltmän, ilki 5 –e köpeldýäris we soňra 9-a bölýäris. Beýle etmegimiziň sebäbi başgada köp dillerde bolşy ýaly C dilinde hem bitin sanlar bölünende kesilmegi bilen, ýagny drob böleginiň taşlanýanlygy bilen baglanyşyklysy. Şeýlelikde 5 we 9 bitin sanlar, kesilme netijesinde 9/5 aňlatma nul bahany kabul eder, netijede Selsiý boýunça temperaturanyň ýerine nul baha çap edilirdi.

Bu mysal printf – iň işleýşi barada hem düşüňjämizi giňeltdi. printf – munuň özi formatly giriş-çykyşyň uniwersal funksiýasydyr, ol barada 7-nji başda giňişleýin beýan ediljekdir. Onuň birinji argumenti string, ondaky her bir % bellik bolsa indiki argumentler(ikinli, üçünji) bilen baglanyşykly bolup, % belgisiniň yzyndan gelýan maglumat bolsa olaryň her-biriniň çapa çykjak görnüşini

kesgitleýär. Mysal üçin %d argumentiň bitin san görnüşde çapa çykarylmalgyny gurnayär,

```
printf("%d\t%d\n",fahr, celsius);
```

instruksiýa bolsa bitin fahr ululygy çapa çykaryär, tabulýa goýýar we bitin celsius ululugu çapa çykaryär.

printf functionynda birinji argumentiň her bir spesifikatoryna(% bilen başlanýan gurluş) ikinji, üçünji, we ş. m. gabat gelyär. Spesifikatorlar we olara degişli argumentler sany we tipi boýunça gabat gelmelidirler: eger gabat gelmese onda maglumat nädogry görnüşde çapa çykarylýar.

Ýeri gelende aýtsak printf C diliniň bölegi däl, umuman C dilinde giriş-çykyşa kesgitleýän ýörite konstruksiýa ýokdur. printf functiony C programma elýeterli bolan standart bibliotekadaky peýdaly functiondyr. printf functionynyň häsiýeti ANSI standart tarapyndan kesgitlenendir we standarta tabyn bolan hemme C sistemalarda birmeňzeşdir.

C-iň özüne ünsüňizi jemlemek üçin giriş-çykyşa barada 7-nji baba çenli köp gürrüň etmeris. Şeýle hem ady agzalan baba çenli formatly giriş barada-da gürrüň etmeris. Egerde sizde sanlary girizmek zerurlygy çyksa, onda **scanf** functionyna degişli 7.4 bölümi okamaklygy maslahat berýäris. Bu functionyň ýazylyşynda printf bilen meňzeşlik bar, ýöne ol maglumatlary çapa çykarmaga däl-de girizmekligi üpjün edýär.

Temperaturany özgertmek programmasy bilen baglanyşykly ýene iki sany düzedilmeli mesele bar. Olaryň birinjisi(has ýönekeýi) çapa çylarylýan sanlar gysyk, ýagny olar kolonkanyň sag pozisiýasy boýunça deňlenmändir. Onu düzetmek üçin, her bir %d spesifikatora oňa berilýän meýdançanyň uzynlygyny görkezýän formaty goşmak ýeterlikdir; Şondan soňra programma sanlary görkezilen meýdançanyň sag tarapyna gysmak bilen çapa çykarar. Mysal üçin

```
printf("%3d %6\n", fahr, celsius);
```

görnüşde ýazsak, onda her bir setirde birinji san üç pozisiýada, ikinji san bolsa alty pozisiýada ýerleşdiriler. Netijede aşakdaky ýaly görnüşde çapa çykarylýar:

```
0      -17
20     -6
40      4
60     15
80     26
100    37
...
```

Düzedilmeli meseleleriň ikinji has zerury biz bitin sanly arifmetikany ulanp hasaplama geçirdik, şonuň üçin hem Selsiý boýunça temperaturanyň tablisasy beýle bir takyk däl. Mysal üçin 0°F hakykatda(onluk takyklykda) -17,8°C-a dälde -17-ä deň. Has takyk temperaturanyň bahasyny hasaplamamak ucun biz bitin sanly däl-de eýsem ýüzýän nokatly arifmetikadan peýdalanmaladyrys. Bu bolsa programmada birnäçe üýtgetmäni geçirmekligi talap edýär.

```
#include <stdio.h>
```

```
/* fahr = 0, 20, ..., 300 üçin temperaturalaryň Ferengeýt we Selsiý
```

```

        boýunça tablisyny çap etmek */
main()
{
    float fahr, fahr, celsius;
    int lower, upper, step;

    lower=0;
    upper=300;
    step=20;

    fahr = lower;
    while (fahr <= upper) {
        celsius = (5.9/9.0) * (fahr-32.0);
        printf("%3.0f %6.1f\n", fahr, celsius);
        fahr = fahr + step;
    }
}

```

Programmanyň görnüşi azajyk üýtgedi. Ol öňki programmadan diňe fahr we celsius üýtgeýän ululyklaryň float tipe degişli edilip beýan edilenligi bilen hem-de öwürmek formulalarynyň has tebigylygy bilen tapawutlanýar. Öňki programmada 5/9 ýazgyny ýazyp bolmaýardy, sebäbi bitinsanlaýyn bölmäniň netijesi nula deň bolardy. Konstantadaky onluk nokat onuň ýüzýän nokatly san hökmünde seredilmegine sebäp bolýar we şeýlelikde 5.0/9.0 ýazgy iki sany ýüzýän nokatly sanyň gatnaşygy ýaly seredilýär we onuň drob böleginiň taşlanmasy bolmaýar. Arifmeiki operasiýa bitin operandlara eýe bolanda ol bitinsanly arifmetikanyň düzgünleri boýunça ýetýär. Eger bir operand ýüzýän nokatly san bolup, beýlekisi bitin san bolsa, onda operasiýa ýerine ýetmezden öň bitin tipli bölegi ýüzýän nokatly tipe öwürüler. Eger biz fahr-32 görnüşde ýazan bolsak, onda 32 san awtomatik usulda ýüzýän nokatly sana öwürülerdi. Şeýlede biz ýüzýän nokatly konstantany ýazanymyzda elmydama onluk nokady ulanarys, hatda konstanta bitin tipe degişli bolan ýagdaýynda hem. Munuň özi programmany okaýan adamynyň programmanyň tebigatyna oňat düşünmegine ýardam edýär.

Haýsy ýagdaýlarda bitin ululyklaryň ýüzýän nokatly ululyklara öwrülýänligi 2-nji bapda düýpli seredilýär. Häzir bolsa bir ýagdaýa üns breliň, aşakdaky eýe bolmak

```
fahr = lower;
```

we şerti barlamak

```
while (fahr <= upper)
```

tebigy şekilde işleýär, ýagny operasiýa ýerine ýetmezden öň **int** tipe degişli baha **float** baha öwürülýär.

printf functionynda %3.0f spesifikasiýa ýüzýän nokatly sanyň(häzirki ýagdaýda fahr ululygyň) giňligi üç pozisiýadan uly bolmadyk meýdançada, onluk

nokatsyz we drob böleksiz çapa çykarylýanlygyny aňladýar. %6,1f spesifikasiýa beýlki bir sanyň(celsius) giňligi alty pozisiýaly meýdançada onluk nokatdan soň bir san bilen çapa çykarylýanlygyny aňladýar. Aşäkdakylar çapa çykarylar:

0	-17.8
20	-6.7
40	4.4

Meýdançanyň giňligini we takyklygy görkezmän hem bolar: %6f sanyň alty pozisiýadan köp ýer almajakdygyny aňladýar; %2f – san onluk nokatdan soň iki bahany alýar, ýöne giňligi çäklendirilmeyär; %f ýöne ýüzýän nokatly sanyň çapa çykarylýanlygyny görkezýär.

%d onluk bitin sany çap etmek

%6d 6 pozisiýaly meýdançanyň içinde onluk sany çap etmek

%f ýüzýän nokatly sany çap etmek

%6f ýüzýän nokatly sany 6 pozisiýaly meýdançada çap etmek

%.2f ýüzýän nokatly sany onluk nokatdan soň 2 sifr bilen çapa çykarmak

%6.2f ýüzýän nokatly sany 6 pozisiýaly meýdançanyň içinde we onluk nokatdan soň 2 sifr bilen çapa çykarmak.

Mundan başgada **printf** functionynda indiki spesifikatorlary ulanmaklyga ýol berilýar: sekizlik san üçin %o, onaltlyk san üçin %x, simwol(litera) üçin %c, string(setir) üçin %s we % simwolyň özi üçin %%.

Ýumuş 1. Tablisanyň ýokarsynda sözbaşy çap ediler ýaly edip, temperaturany özgertmek programmasyny kämilleşdiriň

Ýumuş 2. Ferengeýtdäki temperaturanyň Selsiýdäki temperatura bilen baglanyşyk tablisasyny çapa çykarýan programma düzüň.

for operatory

Şol bir programmany ýazmaklygyň köp dürli usullary bar. Biziň temperaturany özgertmek programmamyzyň görnüşini üýtgedeliň:

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    int fahr;
```

```
    for (fahr=0; fahr <= 300; fahr=fahr+20)
```

```
        printf(“%3d %6.1f\n”, fahr, (5.0/9.0)*(fahr-32));
```

```
}
```

Bu programma hem şol bir netijäni çapa çykarýar, ýöne görnüşi gürrüňsiz başga hilidir. Esasy tapawuplanýan ýeri köp üýtgeýän ululyklaryň ýoklugyndadyr. Diňe int tipli **fahr** üýtgeýän ululygy bar. Biziň üçin täze bolan **for** instruksiýasynda **fahr** üýtgeýän ululygynyň başlangyç we ahyrky bahalary, şeýle hem onyň uüýtgeýiş ädimi konstanta görnüşinde ýazylypdyr, a Selsiý boýunça temperaturany hasaplaýan aňlatma bolsa aýratyn eýe bolmak instruksiýasy görnüşinde däl-de **printf** functionynyň üçünji argumenti bilen berlipdir.

Soňky üýtgetme aşakdaky umumy düzgüniň ulanylyşyna mysal bolup biler: islendik tipe degişli üýtgeýän ululyga derek şol tipe degişli has çylşyrymly aňlatma

hem ulanylyp biliner. Şeýlelikde **printf** functionynyň üçünji argumentiniň ýerine %6.1 spesifikatoryň talabyna görä süýşýän nokatly ululyk bolmaly, şoňa görä-de bu ýerde şol tipe degişli islendik aňlatma bolup biler.

for Instruksiýasy **while** gaýtalanmasynyň umumylaşdyrylan görnüşini beýan edýär. Eger siz ony öň ýazylan while bilen dňeşdirseňiz onda onuň işleýişi aýdyň bolar. Ýaýlaryň içinde biri-birinden nokatlar bilen bölünen üç sany aňlatma bar. Birinji aňlatma – inisalizasiýa

fahr=0

-gaýtalanmaklyga girmezden öň bir gezek ýerine ýetýär. Ikinji – gaýtalanmaklygyň dowam etmeklik şertiniň barlagy

fahr<=300

Şert hasaplanylýar, eger ol **çyn** bolsa onda gaýtalanmaklygyň göwresi(biziň mysalymyzda printf-e bir gezek ýüzlenmek) gaýtalanýar. Ondan soňra indiki ädime geçilýär, bir ädim artdyrylýar:

fahr=fahr+20

we täzedan şert barlanylýar. Haçanda şert **ýalan** bolanda gaýtalanmaklyk tamamlanylýar. while ýaly for gaýtalanmasynyň göwresi bir instruksiýadan ýa-da figuraly ýaýlaryň içine alnan birnäçe instruksiýadan ybarat bolup biler. Bu ýerdäki üç aňlatmanyň(inisializasiýa, şert we ädimiň ulalmagy) ýerinde islendik aňlatma bolup biler.

Programmanyň düşnükli bolmagy nukdaý-nazaryndan seredilip while ýa-da for saýlanylyp alnyp biliner. Haçanda inisilizasiýa hem-de ädim artmasy şol bir ululyk arkaly logiki baglanyşykly bolsa we bir instruksiýa arkaly aňladylýan bolsa onda for has amatly bolýar, sebäbi ol while - dan tygşytly ýazylýar, onuň dolandyryjy bölegi bir ýere jemlenendir.

Ýumuş 1. *Temperaturany özgerdýän programmany, tablisany ters tertipde çap eder ýäly(ýagny 300 –den 0 –a çenli) özgerdiň.*

Atlandyrylan konstantalar(hemişelikler)

Temperaturany özgertmek programmasyny tamamlamazdan öň ýene bir pikir aýdalyň. Haçanda programmada 300, 20 ýaly “uly sanlar” dagaşyp ýatan bolsalar bu örän erbet hasaplanylýar. Programmany okajak adam üçin ol sanlaryň hakykatda näme aňladýanlygy barada hiç-hili ýaňzytma hem ýok. Mundan başgada olary haýsydyr bir sistematiiki usulda başgalara çalyşmak kyndyr. Beýle sanlar bilen işlemekligi ýeňilleşdirmekligiň bir mümkin bolan usuly olara manyly at dakmakdyr. #define setiri berlen setir üçin *simwiolly ady* ýa-da *atlandyrylan konstantany* kesgitleýär:

#define at goýulýan tekst

Şu pursatdan başlap *at* ýüze çykan ýerinde(eger ol goşa dyrnagynyň içinde ýazylan setirde we beýleki bir adyň bölegi bolmadyk ýagdaýynda) ol *goýulýan tekst* bilen çalşyrylar. *at* hem üýtgeýän ululyk ýaly görnüşe eýedir: harpdan başlanýan harplaryň we sanlaryň yzygiderligi. *goýulýan tekst* simwollaryň islendik yzygiderligi bolup biler, özünem olaryň diňe sandan ybarat bolmagy hökman däl.

```
#include <stdio.h>
#define LOWER 0      /* tablisanyň aşaky çägi */
#define UPPER 300    /* ýokarky çägi */
#define STEP 20      /* ädimiň ölçegi */
/* Ferengeýt we Selsiý boýunça temperatura tablisasyny çap etmek */
main()
{
    int fahr;
    for (fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));
}
```

LOWER, UPPER, we STEP ululyklar üýtgeýän ululyklar bolman eýsem atlandyrylan konstantalardyr. Şonuň üçin olar beýan edilmeyärler. Kabul edilen umumy ylalaşyk boýunça, atlandyrylan konstantalaryň atlary kiçi harp bilen ýazylýän adaty üýtgeýän ululyklardan tapawutlylykda, baş harp bilen ýazylýär. #define – setiriň yzynda nokatly oturyň bolmaýanlygyny hem belläp geçeliň.

Literany çapa çykarmak

Indi biz tekstler bilen işlemäge mümkinçilik berýän programmalaryň toplumyna seretmelçi. Siz durmuşda ulanylýan programmalaryň köpüsiniň şu programmalaryň ösdürilen nusgalarydygyna göz ýetirersiňiz.

Standart kitaphana giriş-çykyşyň örän ýönekeý modelini goldaýar. Tekst giriş-çykyş nireden gelýänligine ýa-da nirä barýanlygyna bagly bolmazdan literanyň akymy bilen iş salyşýar. *Tekst akymy* – munuň özi her biri nul ýa-da ondan köp litera saklaýan we täze setir literasy bilen tamamlanýan setirlerden ybarat bolan literalaryň yzygiderligidir. Islendik giriş-çykyş akymynyň bu modele gabat gelmekligine göz-gulak bolmak jogapkärçiligi bibliotekanyň üstüne atylandyr: programmist bibliotekany ulananda programmanyň daşynda ol setirleriň nähili görnüşiniň bardygy barada aladalanmaly däldir.

Standart biblioteka bir literany okamak we ýazmak üçin birnäçe functiony özünde saklaýar. Olaryň in ýönekeýi – **getchar** we **putchar**. getchar functionyna bir gezek ýüzlenilende, tekst akymdan netije görnüşde berilýän *indiki litera* okalýar. Şeýlelikde

```
c = getchar();
```

ýerine ýeteninden soň c üýtgeýän ululyk girizilýän indiki literany kabul edýär. Adatça litera klawiaturadan girizilýär. Faýldan girizilmek 7-nji bapda serediiýär.

Putchar functiony ýüzlenilende ol bir literany çapa çykarýär. Ýagny,

```
putchar();
```

c bitin tipli üýtgeýän ululygyň bahasyny litera görnüşde çapa çykarýar (adatça ekrana). putchar we getchar functionlaryň çagyrylyşy islendik nusgada gezekleşip gelip bilerler. Bu functionlara ýüzlenilişi ýaly tertipde hem çapa çykarmaklyk emele geler.

Faýlyň kopiýasyny almak

getchar we **putchar** funtionlaryň barlygynda, giriş-çykyş barada başga hiç zat bilmäýeniňde-de, örän täsin kän peýdaly programmalary ýazmak bolar. Ýönekeý mysal – bir literadan giriş akymdan çykyş akyma kopiýasyny alýan programma:

```
literany okamak
while (litera faýlyň soňyny görkezýän litera bolmasa)
    ýaňky okalan literany çapa çykarmaly
literany okamak
```

ýokardaky ýazgylary C dilindäki programma görnüşinde beýan edeliň:

```
#include <stdio.h>
/* girişi çikişa kopiýa etmek; 1-nji görnüş */
main()
{
    int c;
    c = getchar();
    while (c != EOF) {
        putchar(c);
        c = getchar();
    }
}
```

Gatnaşyk operasiýasy bolan != “deň däldir” – diýmekligi aňladýar.

Klawiaturadan girizilýän, ekrana çykarylýan ýa-da maşynyň içindäki her bir litera bitleriň birleşmesi görnüşinde kodirlenýär. Char tip, ýörite liter maglumatlary saklamaklyk üçin niýetlenendir, ýöne munuň üçin islendik bitin tipi ulanmak bolar. Ýöne biz düşündirilmegi zerur bolan, möhüm bir sebäbe görä int tipi peýdalanýars.

Şeýle kynçylyk ýüze çykýar: girişiň soňyny adaty okalýan maglumatlardan nähili tapawutlandyrmaly. Bu meseläniň esasy çözüwi, getchar functiony giriş akymy tamamlananda hakyky literalaryň hiç biri bilen meňzeş bolmadyk bahany gaýdyp bermelidir. Ol baha EOF-a(end of file – faýlyň soňy – diýen sözleriň baş harplary) deňdir. Biziň ulanýan tipimiz, getchar functionynyň berip bilýän hemme bahasyňa “ýeter” ýaly derejede bolmalydyr. Biz bu ýagdaýda char tipini ulanyp bilmeýäris, sebäbi **c** ululyk char tipiň hemme bahalaryndan başga ýene-de EOF-y saklap bilmelidir. Ine şonuň üçin hem biz char tipini däl-de int tipini ulandyk.

EOF <stdio.h>-da kesgitlenen bitin hemişelikdir. Onuň nähili baha alýanlygy möhüm zat däldir, esasy zat onuň bahasy char tipiň kabul edýän bahasyndan tapawutlanmalydyr. Aýratyn atly atlandyrylan hemişeligiň ulanylmagy, programmanyň takyk san baha bagly bolmazlygyna kepil geçýär, ol ululyklar beýleki C ulgamlardakydan mümkingadar tapawutlanar.

Kopiýasyny alýan programmany has gysga hem ýazyp bolar. C dilinde her bir eýe bolmak, mysal üçin

```
c = getchar()
```

çep gapdaldaky ululyga dakylan, belli bir baha deň bolan aňlatma hökmünde seredilýär. Munuň özi has çylşyrymly aňlatmalaryň içinde hem eýe bolmaklygyň gabat gelip biljekdigini aňladýar. Eger eýe bolmaklygy while gaýtalanmanyň şert barlaýan böleginde ýerleşdirsek, onda faýlyň kopiýasyny alýan programmany aşakdaky ýaly ýazmak bolar:

```
#include <stdio.h>
/* girişi çikişa kopiýa etmek; 2-nji görnüş */
main()
{
    int c;
    while ((c = getchar()) != EOF)
        putchar();
}
```

while gaýtalanmasy getchar-yň netijesini c ululyga dakýar we şol wagt barlaýar: ol ululyk “faýlyň soňymy” ýagny EOF-a deňmi. Eger deň däl bolsa, onda while gaýtalanmasy ýerine ýetýär we litera çapa çykarylýar. Giriş akymynyň tamamlanmagy bilen, while öz işini tamamlayar, şonlukda main hem öz işini tamamlayar.

Programmanyň bu görnüşinde getchar-a diňe bir gezek ýüzlenilýär, ýagny girizmek “merkezleşdirilen”. Şonuň netijesinde ol has tygşytly we ýeňil okalýar. Eýe bolmaklygyň şert barlamak bilen gabat gelýän ýagdaýlary size ýygy-ýygydan gabat geler.(Bu ýagdaýa aşa ýykgyň etmeklik, programmany okamaklygy kynlaýar, şonuň üçin bu mümkinçiligi akyly-başly ulanmak gerek).

Şertiň içinde eýe bolmaklygy ýaý şekilli skobkanyň içinde ýazmak zerurdyr. != operasiýanyň möhümligi = operasiýanyň möhümliginden ýokarydyr, şonuň üçin skobkalaryň bolmadyk ýagdaýynda != operasiýa eýe bolmak operasiýasyna çenli ýerine ýeter. Şonuň üçin hem

```
c = getchar() != EOF
```

ýazgy

```
c = (getchar() != EOF)
```

ýazga ekwiwalentdir. A bu bize düýünden gerek zat däl: c üýtgeýän ululyga getchar functionynyň faýlyň soňy alamatyna düşýanlygyna baglylykda 0 ýa-da 1 dakylar.(Bu barada 2-nji bapda has giňişleýin aýdylar).

Ýumuş 1. getchar() != EOF aňlatmanyň 0 ýa-da 1 bahany alýandygyny barlap göz ýetiriň.

Ýumuş 2. EOF-iň bahasyny çapa çykarýan programma düzüň.

Literi sanamak

Aşakdaky programma literi sanamak bilen meşgullanýar; onuň kopiýa alýan programma bilen köp meňzeşlikleri bar.

```
#include <stdio.h>
/* girizilýän literi sanamak; 1-nji görnüş */
main()
{
    long nc;
```

```

nc = 0;
while (getchar() != EOF)
    ++nc;
printf("%ld\n",nc);
}

```

instruksiya ++nc, täze ++ operatory görkezýär, ol *bir birlik artdyrmaklygy* aňladýar. Onuň ýerine nc = nc + 1 hem ýazyp bolar, ýöne ++nc has gysga, a köp ýagdaýda effektiv (beýlekä garanda, peýdaly). Edil şonuň ýaly *bir birlik kemeltýän* -- operator hem bar. ++ we - - operatorlar öňünden(++nc) we soňundan(nc++) gelýän görnüşlerde bolup bilerler. Bu iki görnüş aňlatmalarda dürli baha eýe bolýanlygy, ýöne ++nc we nc++ ikisiniň hem nc ululyga bir birlik goşýalygy 2-nji bapda görkeziler. Häzirki ýagdaýda biz öňünden gelýän ýazgynyň üstünde durup geçeris.

Liter sanaýan programma sanalan literleriň jemini long tipli ululykda saklaýar. Bitin tipli long 32 bitden az däl. Käbir maşynlarda int we long tipleri birmeňzeş ölçege eýe bolsalarda, int tipi 16 bit saklaýan maşynlar hem bar. Olaryň maksimal kabul edip bilýän bahasy 32767, bu beýle bir uly san däl, şonuň üçin hem ol tipe degişli bolan üýtgeýän ululyklaryň dolmagy (переполнение) mümkindir. printf functionynda %ld öwürmek spesifikasiýasy long tipli argumente gabat gelýär.

Eger double(ikeldilen takyklykdaky float tipi) tipi ulanmak bilen has uly aralygy hem almak bolar. Şeýle hem gaýtalanmaklygyň beýleki usulyny görkezmek üçin while derek for ulanlyk

```

#include <stdio.h>
/* girizilýän literi sanamak; 1-nji görnüş */
main()
{
    double nc;
    for (nc=0; getchar() != EOF; ++nc)
        ;
    printf("%.0f\n", nc);
}

```

printf –de %f speaifikator float tipli üýtgeýän ululyklar üçin ulanylýar, ýagny häzirki ýagdaýda double üçin; speaifikator %.0f onluk nokatsyz we drob böleksiz çapa çykarmak (bu mysalda drob san alynmaýar).

Bu ýerde **for** gaýtalanmasynyň göwresi boş, ýagny şert barlamakdan we sanaýjyny artdyrmakdan başga hiç zat etmek gerek däl. Ýöne C diliniň grammatikasynyň düzgüni boýunça for gaýtalanmasynyň göwresi bolmalydyr. Bu talaby, boş instruksiya diýilip atlandyrylýan izolirlenen nokatly otur ýerine ýetirýär. Biz has düşnükli bolmagy üçin nokatly otury aýratyn setirde goýdyk.

Iň soňunda belläp geçmeli, egerde giriş özünde bir literany hem saklamaýan bolsa, getchar-a birinji gezek ýüzlenilende while ýa-da for gaýtalanmalarynda şert ýerine ýetmez we programmanyň netijesi nul bolar, munuň özi dogry netijedir. Muňa düşünmek möhümdir. while we for gaýtalanmaklarynyň ajaýyp häsiýetleriniň biri şert gaýtalanmaklygyň göwresi ýerine ýetmezden öň

barlanylýar. Hiç zat etmeli däl bolsa, hiç zat hem edilmez, hatda şerte baglylykda gaýtalanmaklygyň göwresi bir gezek hem ýerine ýetmän biler. Programma girizilýän literalaryň sany nula deň bolan ýagdaýynda hem özüni dogry(корректно) alyp barmalydyr. while we for gaýtalanmaklarynyň gurluşlary şertiň çäklerinde programmanyň özüni dogry alyp barjaklygyna goşmaça ynam döredýär.

Setiri sanamak

Aşakdaky programma setiri sanaýar. Ýokarda ýatlanylşy ýaly, standart biblioteka her bir setiri täze setir literasy(\n) bilen tamamlanýan setirlerden düzülen tekst akymynyň giriş-çykyş modeli bilen üpjün edýär. Şeýlelikde setiri sanamak täze setir(\n) literasyny sanamaklyga syrygýar.

```
#include <stdio.h>
/* giriş akymyň setirlerini sanamak */
main()
{
    int c,nl;
    nl=0;
    while ((c = getchar()) != EOF)
        if (c == '\n')
            ++nl;
    printf("%d\n", nl);
}
```

Indi gaýtalanmaklygyň göwresini, garamagynda nl sanaýjynyň bahasyny bir birlik ulaltmaklyk ýerleşýän **if** instruksiya emele getirýär. if instruksiya skobkanyň içindäki şerti barlaýar we eger ol çyn bolsa onda onuň yzyndan gelýän instruksiya (ýa-da figuraly skobkanyň içinde ýerleşen instruksiýalaryň toplumy) ýerine ýetýär. Biz ýene-de programmanyň dolandyrylyşynyň düşnükli bolmagy üçin boşluplary ulandyk (haýsy instruksiya haýsy instruksiýany dolandyrýar).

C dilinde ikileýin deňdir alamaty(==) “deňdir” operatoryny aňladýar (Pascaldaky = we Fortrandaky .EQ.). Deňligi barlanylýan operatora Ikileýin deňdir alamatynyň(==) ulanylmagynyň sebäbi ony bir deňdir ulanylýan eýe bolmak operatoryndan tapawutlandyrmak üçindir. Duýdyrýarys: C dilinde programma düzmeklige ýaňy girişenler = ýazmak bilen == göz önünde tutýarlar. Beýle ýagdaýda 2-nji bapda ýazylyşy ýaly, aňlatma görnüşi boýunça ýolbererlikdir, kompilyator hiç-hili duýdyryjy habar bermeýär.

Bir dynagyň içinde ýazylan litera, şol literanyň kodyna deň bolan bitin sany görkezýär(berlen maşynda kabul edilen kodirowka görä). Bu *liter hemişeligi* diýilip atlandyrylýar. Kiçi bitin sanlary ýazmagyň başga bir usuly hem bar. Meselem ‘A’ liter hemişelikdir; ASCII liter ýygynyndasynda onuň bahasy 65-e deňdir, ýagny 65 A harpynyň maşynyň ýadynda bellenilişidir. Elbetde ‘A’ konstanta görnüşinde 65-den göze ýakymlydyr, yagny birinji ýazgynyň manysy has düşnüklidir, ol literalaryň kodirleniş usulyna bagly dälendir.

String(setir) hemişeliklerinde ulanylýan Esc-yzygiderlikleri liter hemişeliklerinde hem ulanmak bolýar. Ýagny ‘\n’ täze setir literasynyň kody, ol ASCII –de 10 deňdir. ‘\n’ bir litera hökmünde seredilýänligine aýratyn ünsi çekmelidir(onuň kody aňlatmada bitin san görnüşinde seredilýär), “\n” ululyga bolsa setir hemişeligidir. Setir we liter hemişelikleriniň tapawudyna 2-nji bapda düýpli seredilip geçiler.

Ýumuş 1. Boşlugy, tabulýasiýany we täze setiri sanamak üçin programma düzüň.

Ýumuş 2. Girizilýän literalary, yzly-yzyna gelýän boşluklary bir boşluk bilen çalyşmak arkaly çykyş akymyna kopirleýän programma düzmeli.

Ýumuş 3. Girizilýän literalary: tabulýasiýa literasyny \t-a, backspace literasyny \b-a we ters ýapgyt çyzygy \\-a çalşyryp çykyş akymyna kopirleýän programma ýazmaly. Bu hemme tabulýasiýa we backspace literalaryny “görkeklige” mümkinçilik berer.

Söz sanamak

Biziň peýdaly programmalarymyzyň toplumynyň dördünjisi setirleri, sözleri we literalary sanaýan programma bolup, bu ýerde söz diýilip özünde boşluklary, tabulýasiýalary we “täze setir”-leri saklamaýan stringe düşünilýär. Bu programma UNIX sistemasyndaky wc programmanyň sadalaşdyrylan görnüşidir.

```
#include <stdio.h>
#define IN 1 /* sözüň içi */
#define OUT 0 /* sözüň daşynda */
/* setirleri, sözi we literleri sanamak */
main()
{
    int c, nl, nw, nc, state;
    state = OUT;
    nl = nw = nc = 0;
    while ((c = getchar()) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

Programma her gezek sözüň sag literasynyna gabat gelende, sözi sanaýan ululygyň bahasyny bir san artdyryar. state üýtgeýän ululygy sözüň içinde ýa-da daşynda ýerleýändigimizi kesgitleýär. Ilki başda ol ululyga OUT baha dakylýar, ol

“sözün daşynda” daşyndadygyny aňladýar. Biz programmanyň has düşnükli bolmagy üçin 1 we 0 bahalaryň özünden dälde IN we OUT atlandyrylan konstantalary ulanmaklygy maslahat berýäris. Biziň bu kiçijik programmamyzda bu usulyň beýän ullakan zady ýok hem bolsa, uly programmalar ýazylanda onuň düşnükliiliginiň ýokarlanmagy, goşmaça sarp edilen kiçijik zähmediň ýerine düşdügi bolar. Siz programmada ulanylýan uly sanlar, atlandyrylan konstantalar görnüşinde ýazylan ýagdaýynda, programma uly üýtgeşmeleriň girizmekligiň ýeňil bolýandygyny görersiňiz.

```
nl = nw = nc = 0;
```

setir üç üýtgeýän ululygyň bahasyna nul ýazýar. Bu ýazga üňs berip seretsek ol üýtgeşik bir zat dälde we ulanarlykdyr, ýagny eýe bolmak özüniň hususy bahasy bolan aňlatmadyr we eýe bolmak operasiýasy sagdan çep tarap ýerine ýetýär. Ol setir aşakdaky setire ekwiwalentdir:

```
nl = (nw = (nc = 0));
```

|| - operatory ÝADA atly logiki operasiýany aňladýar, şeýlelikde aşakdaky setir:

```
if (c == ' ' || c == '\n' || c == '\t')
```

“eger c ululyk boşluk ýa-da “täze setir” ýa-da tabulýasiýa bolsa” – diýilip okalýar. (\t esc-zygiderlik tabulýasiýa literasyny aňladýandygyny yatlalyň.) Şeýle hem logiki WE – ni aňladýan && operator bar. Onuň möhümligi || operatoryňkydan ýokarydyr. && ýada || operatorlar bilen bagly aňlatmalar çepden saga hasaplanýar, özünem şertiň çynlygy ýada ýalanlygy belli bolan halatynda hasaplamanýň tagtadyljakdygyna güwä geçilýär. Eger c ululyk boşluk bolsa onda c ululygyň “täze setir”-e ýada tabulýasiýa deňdigini barlamak gerek däl. Bölek ýagdaýlarda beýle usulda hasaplamak beýle bir möhüm däl, ýöne has çylşyrymly ýagdaýlarda ony ulanmaklygyň manysy bardyr, biz onyň bilen gelejekde iş salşarys.

Ýokarky mysalda *else* sözi hem gabat geldi, ol *if* –däki şert ýerine ýetmedik ýagdaýynda ýerine ýetirmeli instruksiýany görkezýär. Umumy görnüşde şertli instruksiýa şeýle ýazylýar:

```
if (aňlatma)
```

```
    instruksiýa1
```

```
else
```

```
    instruksiýa2
```

if-else konstruksiýasynda görkezilen instruksiýalaryň diňe biri ýerine ýetýär. Eger *aňlatma* **çyn** bolsa, onda *instruksiýa1*, eger **ýalan** bolsa onda *instruksiýa2* ýerine ýetýär. Bu instruksiýalaryň her biri ýaha bir instruksiýa, ýa-da figuraly skobkalaryň içinde ýazylan birnäçe instruksiýalardan ybaratdyr. Biziň programmamyzda else-den soň iki sany şonuň ýaly instruksiýany dolandyryýan if instruksiýa bar.

Ýumuş 1. Söz sanaýan programmany nähili testirlemeli? Nähili ýalňyşlygy bu programmanyň duýmazlygy mümkin.

Ýumuş 2. Öziniň girişindäki sözleriň her birini aýratyn setirde çapa çykarýan programma düzüň.

Massiwler

A indi bolsa aýratynlykda her-bir sifri, boşluk literalaryny(boşluk, tabulýasiýa we täze setir) we beýleki literalary sanaýan programma ýazalyň. Bu programmanyň ulanyljak ýeri az hem bolsa, ol bize bir mysalda C diliniň ýene-de birnäçe mümkinçiliklerini görkezýär. Girizilýä literalaryň on iki kategoriýasy bar. On sany sifri sanaýjylary on sany aýratyn ululyklarda saklaman, bir massiwde ýerleşdirmek amatly. Ol programmanyň bir görnüşi:

```
#include <stdio.h>
/* sifrleri, boşluk we beýleki literalary sanamak */
main()
{
    int c, i, nwhite, nother;
    int ndigit[10];
    nwhite = nother = 0;
    for (i=0; i < 10; ++i)
        ndigit[i] = 0;
    while ((c = getchar()) != EOF)
        if (c >= '0' && c <= '9')
            ++ndigit[c-'0'];
        else if (c == ' ' || c == '\n' || c == '\t')
            ++nwhite;
        else
            ++nother;
    printf("sanlar=");
    for (i = 0; i < 10; ++i)
        printf(" %d", ndigit[i]);
    printf(", boşluklar = %d, beýlekiler = %d\n", nwhite, nother);
}
```

Eger bu programmany işe goýbersek hem-de programmanyň tekstini giriş hökmünde girizsek, onda aşakdaky netije alnar:

sanlar = 9 3 0 0 0 0 0 0 1, boşluklar = 141, beýlekiler = 347

```
int ndigit[10]
```

deklarasiýa 10 bahadan ybarat, int tipli massiwi kesgitleýär. C dilinde massiwiň elementleri elmydama nuldan başlap belgilenýär, şeýlelikde bu massiwiň elementleri ndigit[0], ndigit[1], ..., ndigit[9] görnüşde bolar, bu ýagdaýy for – gaýtalanmada(inisialisasiýada we massiw çapa çykarylanda) göz önünde tutulmalydyr.

Bitin üýtgeýän ululyklardan(mysal üçin i) we bitin konstantalardan emele gelen aňlatmalar indeks bolup hyzmat edip biler.

Bu programma sifrleriň kodirowkasynyň kesgitli häsiýetine esaslanýar. Mysal üçin

```
if (c >= '0' && c <= '9') . . .
```

barlag, c üýtgeýän ululykda sifriň saklaýanlygyny kesgitleýär. Eger bu şeýle bolsa, onda

```
c - '0'
```

aňlatma sifriň san bahasy bolmaly. Bu aýdylanlar diňe ‘0’, ‘1’, . . . , ‘9’ bahalaryň hatarynda her bir indiki baha öňkiden bir san uly bolan halatynda dogrudyr. Literleriň hemme naborynda bu düzgün saklanylýar.

Kesgitleme boýunça char tipiň bahalary kiçi bitin sanlardyr, ýagny char tipli üýtgeýän ululyklar we hemişelikler arifmetiki aňlatmalarda int tipli bahalara meňzeşdir. Bu tebigydyr we amatlydyr; mysal üçin c-‘0’ aňlatma c ululykdaky 0 – dan 9-a çenli bitin bahany kabul edip bilýär. Şeýlelikde bu aňlatmanyň bahasy ndigit massiwiniň dogry indeksi bolup hyzmat edýär.

Programmanyň aşakdaky bölegi literanyň sanlygyny, boşluk literasydygyny ýa-da başga bir literadygyny kesgitleýär.

```
if (c >= '0' && c <= '9')
    ++ndigit[c-'0'];
else if (c == ' ' || c == '\n' || c == '\t')
    ++nwhite;
else
    ++nother
```

Aşakdaky konstruksiýa

```
if (şert1)
    instruksiýa1
else if (şert2)
    instruksiýa2
...
...
else
    instruksiýan
```

programmada bar bolan birnäçe alternatiw ýoldan birini saýlamak üçin ýygýygydan peýdalanylýar. Şertleriň haýsy hem bolsa biri ýerine ýetýänçä, ýokardan aşak hasaplanylýar; şert ýerine ýetende oňa degişli instruksiýa hem ýerine ýetirilýär we бүтін konstruksiýa öz işini tamamlýar. (Islendik instruksiýa figuraly skobkalaryň içinde ýazylan instruksiýalaryň toplumy bolup biler) Eger şertleriň hiç birisi ýerine ýetmeýän bolsa, onda else sözünüň yzynda ýerleşen iň soňky instruksiýa ýerine ýetýär, (eger şeýle instruksiýa bar bolsa). Eger iň soňky else we instruksiýa ýok bolsa, onda düýbünden hiç zat ýerine ýetmeýär.

Ilkinji if bilen iň soňky else sözünüň arasynda islendikçe mukdarda aşakdaky kombinasiýa bolup biler

```
else if (şert)
    instruksiýa
```

Haçanda olar birnäçe bolsa, onda olary biziň bu ýerde görkezişimiz ýaly formatirlemek amatly bolýar. Eger her bir indiki if sözünü öňündäki else sözüne

göra saga süýşirip ýazsak, onda barlaglaryň juda köp bolan halatynda, tekstiň sahypanyň sag tarapky çäGINE çenli baryp ýetmegi mümkin.

3-nji bapda gürrüňi ediljek switch instruksiýasy C dilinde köpýolly şahalanmanyň gurnamaklygyň başga bir usuly bilen üpjün edýär. Bu usul haçanda geçmek şerti, bitin tipli aňlatmanyň bahasy konstantalaryň berlen toplumynyň birine gabat elen halatynda ulanmak has amatly bolýar. 3.4 bölümde bu programmanyň switch operatoryny ulanmak bilen yazylan görnüşi getirilýär.

Ýumuş 1. Girizilen sözleriň uzynlyklarynyň gistigrammasyny çapa çykarýan programma düzüň. Gistrogrammany kese çyzyklaryň kömegi bilen çekmek ýeňildir. Gistrogrammany dik çyzyklaryň kömegi bilen gurnamak – has kyn mesele.

Ýumuş 2. Girizilen literalaryň gabat geliş ýygylgyny boýunça gistogrammany çapa çykarýan programma düzüň.

Funksiýalar

C dilinde function Fortran dilindäki kömekçi programma we function, Pascal dilindäki procedure we function ýalydyr. Function käbir hasaplama aýratyn ýazmaklyga we onuň gurluşy barada hiç zat bilmezden ony ulanmaklyga mümkinçilik berýär. Function ýazylandan soň, olaryň gurluşy barada hiç zat bilmekligiň geregi ýok, diňe olaryň näme edýänligini bilmek ýeterlikdir. C dilinde functionlary ulanmaklyk amatly, aňsat we peýdalydyr (effektiv). Kähalatlarda siz diňe bir gezek çagyrylýan, gysgajyk functionlara gabat gelersiňiz; olar diňe bir maksat üçin – programmanyň düşnükli bolmagy üçin ýazylandyr.

Biz şu çaka çenli printf, getchar we putchar ýaly taýýar functionlardan peýdalandyk, indi öz hususy birnäçe functionlarymyzy ýazmaklyga wagt boldy. C dilinde Fortran dilindäki ýaly ** derejä götermek operatory ýok. Şonuň üçin bitin m sany bitin položitel n derejä göterýän power(m,n) funkstionynyň mysalynda functionlary kesgitlemeklige seredeliň. power(2,5) functionyň bahasy 32-ä deň. Hakykatda bu functionyň praktiki amiyeti az, ýagny ol diňe kiçi bitin derejeler bilen işläp bilýär, ýöne ol functionyň işleýşine düşünmek üçin oňat mysaldyr. (Standart bibliotekada x^y –ni hasaplaýan pow(x,y) bar.)

Şeýlelikde biziň power we onuň hyzmatyndan peýdalanýan baş functionymyz bar, tutuş programmamyz aşakdaky görnüşde bolar:

```
#include <stdio.h>
int power(int m, int n);
/* power functionynyň testi */
main()
{
    int i;

    for (i = 0; i < 10; ++i)
        printf("%d %d %d\n", i, power(2,i), power(-3,i));
    return 0;
}
```

```

/* base –i n-nji derejä göterýär; n >= 0 */
int power(int base, int n)
{
    int i, p;

    p = 1;
    for (i = 1; i <= n; ++i)
        p = p * base;
    return p;
}

```

Islendik functiony kesgitlemekligiň aşakdaky görnüşi bar:
 netijäň tipi functionyň tipi(parametrleriň toplumy, eger bar bolsa)

```

{
    deklarasiýa
    instruksiýa
}

```

functionlaryň kesgitlenişi islendik tertipde, bir ýa-da birnäçe faýlda (source) bolup biler, ýöne islendik function tutuşlygyna bir faýlda ýerleşmelidir. Eger programmanyň teksti birnäçe faýla bölünen bolsa, onda ony kompilirlmek hem-de ýerine ýetirmek üçin bir faýlda ýerleşdirilen programma garanda köp zatlary ýazmaly bolarys; ýöne bu aýdylanlar, eýýäm programmirleme diline dälde, operasion ulgama degişlidir. Häzirikçe biz bu functionlar bir faýlda ýerleşen diýip hasaplaýarys, şeýlelikde siziň C dilinde ýazylan programmany işletmek üçin öwrenen zatlaryňyz bu ýerde ýeterlik bolar.

Aşakdaky setirde main functionyndan power functionyna iki gezek ýüzlenilýär.

```
printf(“%d %d %d\n”, i, power(2,i), power(-3,i));
```

power functiony her gezek çagyrylanda oňa iki argument berilýär we her gezek baş programma jogap hökmünde bir bitin sany kabul edýär, soňra ol san degişli format boýunça çap edilýär. power(2,i), aňlatmanyň bölegi hökmünde 2 ýa-da i ýaly bitin sany özünde saklaýar. (Hemme functionlaryň netijesi bitin san bolmaýar, bu barada soň, 4-nji bapda aýdylar)

power functionyň birinji setirinde şeýle kesgitleme bar:

```
int power(int base, int n)
```

netijäniň we functionyň hem-de onuň parametriniň tipini görkezýär. Parametrleriň ady powerde lokallaşdyrylandyr, munuň özi islendik beýleki function üçin gizlidir, şonuň üçin beýleki functionlaryň öz maksatlary üçin şeýle atlary ulanmaklyga erkinlik bilýärler. Şol aýdylanlar i,p üýtgeýän ululyklar üçin hem dogrudyr: power functionyndaky i bilen main functionyndaky i ululyklaryň arasynda hiç hili umumylyk ýokdur.

Biz indi function kesgitlenende, skobkalaryň içinde ýerleşdirilen üýtgeýän ululyklara parametr, functiona ýüzlenilende ulanylýan bahalara bolsa argument

diýip atlandyryjakdyys. Biz kä halatda şol bir manyda *formal argument* we *fakt argument* diýen düşüňjeleri ulanjakdyrys.

power functionynyň hasaplamasynyň netijesi, return instruksiýasynyň kömegi bilen main functionyna gaýtarylýar. return sözünüň yzynda islendik aňlatma bolup biler:

```
return aňlatma;
```

Functionyň netije bermegi hökman däl. Eger return instruksiýasynyň aňlatmasy bolmasa, onda ol hiç hili baha gaýtarmazdan, özüni çagyran functiona dolandýrmaklygy geçirýär. Eger hasaplama netijesinde biz functionyň ýapylýan figura skobkaly soňuna ýetsek hem edil ýokardaky ýaly, hiç hili baha gaýtarmazdan, özüni çagyran functiona dolandýrmaklygy geçirýär. Ýüz tutýan functionyň alynýan netijeden ýüz öwürýän ýagdaýlary hem bolup biler.

Siz ähtimal main functionynyň soňundaky return instruksiýasyna üns berensiňiz. Umuman aýdanymyzda main hem beýleki functionlar ýaly function bolup ol öz gezeginde özüne edilen çakyyga jogap bermelidir. Baş function bolan main functiony özüni işe goýberen operasion ulgama netije gaýtarýar. Adatça netije nul bolup, ol işiň normal gutaranlygy barada habar berýär. Eger netije nula deň bolmasa, ol programmanyň adaty däl ýa-da ýalňyş tamamlananlygyny aňladýar. Biz şu çaka çenli programmanyň has düşnükli bolmagy üçin, main functionynda returny ýazmadyk, ýöne indi biz ony ýazarys, sebäbi programma öz tamamlayyş netijesi barada operasion ulgama habar bermelidir.

```
int power(int m, int n);
```

deklarasiýa main functionyň oňünde durmak bilen ol power functionynyň iki sany bitin argumente garaşýanlygyny we bitin tipli netijäni gaýtarýanlygyny aňladýar. *Functionyň prototipi* diýilip atlandyrylýan bu beýan etme power functionynyň hemme çagyryşlary we kesgitlenmesi bilen baglanyşykly bolmaly. Eger functionyň kesgitlenişi ýa-da çagyrylyşy özüniň prototipine gabat gelmeýän bolsa ol ýalňyşlykdyr.

Parametrleriň atlary ylalaşygy talap etmeýär. Umuman olar prototipde islendik bolup biler ýa-da düýbünden bolman hem biler, ýagny prototip aşadaky ýaly hem ýazylyp biliner:

```
int power(int, int);
```

Ýöne öýlanylyp goýulan atlar programmanyň düşnüpli bolmagyny üpjün edýär, şonuň üçin biz köplenç şeýle atlary ulanarys.

Taryhy bellik. ANSI-C diliniň öňki wersiýalardan esasy tapawudy hem functionyň beýan edilişinde we kesgitlenilişindedir. C diliniň ilkinji wersiýasynda power functiony aşadaky görnüşde yazmak talap edilýär:

```
/* power base ululygy n-nji derejä götermek; n >= 0 */
```

```
/* (C diliniň köne stilindäki görnüşi) */
```

```
power (base, n)
```

```
int base, n;
```

```
{
```

```
    int i, p;
```

```
    p = 1;
```

```

        for(i = 1; i <= n; ++i)
            p = p * base;
    return p;
}

```

Bu ýerde parametrleriň atlary ýaý şekilli skobkalarda görkezilýär, a olaryň tipleri bolsa ilkinji açylýan figuraly skobkanyň öňünden berlipdir. Parametriň tipi baradaky görkezme bolmadyk ýagdaýynda, ol int tipine degişli hasap edilýär.(Functioniunyň göwresi üýtgemä sezewar bolmandyr.)

power functionynyň programmanyň başyndaky beýan edilişi C diliniň ilkinji wersiýasyna laýyklykda aşakdaky görnüşde bolamlydyr:

```
int power();
```

Parametrleriň toplumyny görkezmek bolmaýar, şonuň üçin hem kompilýatoryň power functionyna ýüzlenmekligiň dogrulygyny barlamak mümkinçiligi ýok. Ýagny power functionynyň beýan edilişi bolmadyk ýagdaýynda function bitin tipli netijäni berýär diýilip düşünilýär, onda bu ýagdaýda beýan etmekligi doly aýyrmak hem bolar.

Functionlaryň prototipleri üçin täze barlag(sintaksis) argumentleriň sany we olaryň tipi barada ýalňyşlygy kesgitlemekligi ýeňilleşdirýär. Functiony beýan etmekligiň we kesgitlemekligiň köne barlagy ANSI standarty tarapyndak rugsat berilýär, ýöne siziň kompilýatoryňyz täze sintaksisi(barlagy) goldaýan bolsa biz diňe şony ulanmaklygy örän maslahat berýäris.

Ýumuş 1. Temperaturany özgertmekligiň özüni özbaşdak function görnüşinde bölmek arkaly temperaturany özgerdýän programmany täzedan ýazyň.

Argumentler. Bahasy boýunça çagyrmak

C dilindäki functionlaryň bir häsiýeti, beýleki dillerde programma düzmek dillerini ulanýanlar üçin, mysal üçin Fortran dilini ulanýanlar üçin täzelik bolar. C dilinde functionyň hemme argumentleri “bahasy boýunça” geçirilýär. Munuň özi, çagyrylýan functiona argumentleriň özi dälde, argumentleriň bahalary wagtlaýyn ululyklarda geçirilýär. Argumentleri geçirmekligiň beýle usuly argumentleriň lokal kopiýalary bilen dälde, gönüden-göni özi bilen işleýän, Fortrandaky “yuz tutmak boýunça çagyrmak”-dan we Pascal-daky parametrleriň var spesifikasiýasyndan birneme tapawutlanýar.

C diliniň kömekçi programmalary ulanmakdaky esasy tapawudy, çagyrylýan function çagyrylan functionyň üýtgeýän ululygyny(argumentini) üýtgedip bilmeýär: ol diňe onuň bölek, wagtlaýyn kopiýasyny üýtgedip bilýär.

Özünem bahasy boýunça çagyrmaklygy diliň kem tarapy dälde, artykmaç tarapy hasaplamaly. Diliň bu häsiýetine laýyklykda, “keseki” üýtgeýän ululyklarynyň sany az bolan, ýagny parametrlere çagyrylýan kömekçi programmalaryň lokal inisializirlenen ululyklary hökmünde seretmek bilen has tygşytly programma ýazmak bolýar. Bu aýdylanlara mysal edip power functionynyň ýene bir görnüşine seredeliň.

```

/* power base ululygy n-nji derejä götermek; n >= 0; 2-nji görnüş */
int power(int base, int n)

```

```

{
    int p;
    for(p = 1; n > 0; --n)
        p = p * base;
    return p;
}

```

Bu ýerde *n* parametr wagtlaýyn ululyk bolup, ol ululygyň bahasy tä nul bolýança for gaýtalanmasynda kemeldilmek arkaly sanamak amala aşyrylýär. Bu ýagdaýda gaýtalanmany sanamak üçin goşmaça *i* ululygyň zerurlygy ýok. Biz power-ň içinde *n* ululygyň bahasyny üýtgetsek hem ol argumentiň özüne hiç-hili täsir etmez, sebäbi ol function argumentiň kopiýasy bilen çagyrylýär.

Zerurlyk bolan ýagdaýynda çagyrylýan functionyň özüni çagyran programmanyň üýtgeýän ululygyny üýgedip biler etmek mümkinçiligi hem bar. Onuň üçin çagyran programma üýtgedilmäge degişli ululyklaryň adreslerini (salgylaryny, üýtgeýän ululyklaryň görkezijilerini) degişli parametrleri görkeziji hökmünde ulanmak arkaly çagyrylýan functiona geçirmelidir. Görkezijilere degişli zatlaryň hemmesine biz 5-nji bapda seredip geçäris.

Massiwi argument görnüşinde geçirmekligiň mehanizmi bolsa başgaçadyr. Haçanda massiwiň ady functionda argument hökmünde ulanylan ýagdaýynda, onda functiona massiwiň başlangyç adresini görkezýän ululyk geçirilýär; massiwiň hiç-hili elementleri kopirlenmeýär. Function kabul edip alan ululygyny Indeksirlemegiň kömegi bilen massiwiň islendik elementi bilen işläp bilýär. Bu barada indiki bölümde gürrüň ederis.

Literleriň massiwi

C dilinde massiwleriň iň bir ýaýran görnüşi – literleriň massiwidir. Liter massiwleriň we olar bilen işleýän functionlaryň ulanylyşyny görkezmek maksady bilen, tekst setirleriniň toplumyny okaýan hem-de olaryň içinden iň uzynyny çapa çykarýan programma ýazalyň.

Onuň işleýşi diýseň ýönekey:

```

while (ýene-de setir barmy?)
    if (bu setir öňkileriň hemmesinden uzyn)
        ony ýatda saklamaly
        onuň uzynlygyny ýatda saklamaly
    iň uzyn setiri çapa çykarmaly

```

Onuň işleýişinden görnüşi ýaly, programma tebigy görnüşde böleklere bölünýär. Olariň biri täze setiri girizýär, beýlekisi ol setiri barlaýar, üçinji ony ýatda saklaýar, a galanlary bolsa hasaplaýyş prosessini dolandyryar.

Görnüşü ýaly bu iş birnäçe takyk böleklere bölünýär, ol bölekleri şonlugyna hem C diline geçirseň programma düşnükli bolar. Şonuň üçin ilki bilen setiri okamak üçin aýratyn **getline** functionyny ýazalyň. Biz bu functiony beýleki maksatlar üçin hem peýdaly bolar ýaly ýazmaklyga çalyşarys. getline functiony ilki bilen faýlyň(girişiň) soňy barada habar bermelidir, has gowysy ol setiriň uzynlygyny gaýdyp berse (faýlyň uzynlygy nul, haçanda faýlyň soňuna ýetilende).

Nul faýlyň soňuny görkezýän alamat bolup hyzmat edip biler, sebäbi uzynlygy nula deň bolan faýl ýokdyr, hatda diňe bir “täze setir” literasyny saklaýan faýlyň hem uzynlygy 1-e deňdir.

Haçanda biz öňki setirleriň içindäki iň uzyn setirden hem uzyn setire gabat gelsek ony bir ýerlerde ýatda saklamaly bolarys. Bu ýerde şol tapylan setiri ähtibar bir ýere ýerleşdirmek üçin **copy** atly ikinji functionyny ulanarys.

Iň soňunda **getline** we **copy** functionlaryny dolandyrmak üçin baş programma gerek. Biziň programmamyz tutuşlaýyn görnüşde aşakdaky bolar:

```
#include <stdio.h>
#define MAXLINE 1000 /* girizilen setiriň maksimal ölçegi */
int getline(char line[], int maxline);
void copy(char to[], char from[]);
/* iň uzyn setiri çapa çykarmak */
main()
{
    int len; /* alnan setiriň uzynlygy */
    int max; /* barlanylan setirleriň içinden iň uzyny */
    char longest[MAXLINE]; /* alynan setir */
    char longest[MAXLINE]; /* iň uzyn setir */
    max = 0;
    while ((len = getline(line, MAXLINE)) > 0)
        if (len > max) {
            max = len;
            copy(longest, line);
        }
    if (max > 0) /* iň bolmanda bir setir barmy ? */
        printf("%s", longest);
    return 0;
}

/* getline: s setiri okaýar, onuň uzynlygyny gaýtarýar */
int getline(char s[], int lim)
{
    int c, i;
    for (i=0; i<lim-1 && (c=getchar())!=EOF && c!='\n'; ++i)
        s[i] = c;
    if (c == '\n') {
        s[i] = c;
        ++i;
    }
    s[i] = '\0';
    return i;
}
```



```

/* copy: 'from'-dan 'to'-a kopirleýär; to islendik uly bolup biler */
void copy(char to[], char from[])
{
    int i;
    i = 0;
    while ((to[i] = from[i]) != '\n')
        ++i;
}

```

Biz programmanyň başynda beýan edilen **getline** we **copy** functionlary, main bilen bir faýlda ýerleşýär diýip hasaplaýarys.

main we getline functionlary iki sany argument hem-de yzyna gaýtarylýan netije boýunça täsir edişýärler. getline – de argumentler aşakdaky setir bilen kesgitlenýär

```
int getline(char s[], int lim)
```

Görşümüz ýaly onuň birinji argumenti s massiw, a ikinjisi lim bolsa bitin tipli ululyk. Kesgitlenende massiwiň ölçegini görkezmek, ätiýaçdan zerur bolan ýady eýelemek bilen baglanyşyklydyr. getline-yň özünde s massiwiň uzynlygyny görkezmekligiň zerurlygy ýok, sebäbi onuň uzynlygy main-da görleziendir. getline özüne ýüz tutan functiona netijäni gaýtarmak üçin, edil power functionynda edilişi ýaly, return-y ulanýar. Ýokarky setirden görnüşi ýaly getline int tipli bahany yzyna gaýtarýar, egerde functionyň tipi görkezilmedik bolsa onda ol int tipli diýilip hasaplanylýar, şeýlelikde getline functionyň öňünden int sözüni aýyrsak hem bolar.

Käbir functionlar netije berýärler, a käbirleri bolsa hiç-hili netije bermän diňe käbir täsirleri ýerine ýetirýär(copy ýaly). copy functionynda netijäniň tipiniň ýerinde void sözi dur. Bu söz functionyň hic-hili netije bermeýänligini aýdyň görkezýär.

getline functiony döredýän massiwiniň soňunda, onuň soňuny görkezmek maksady bilen '\0' literany ýerleşdirýär. Bu ylalaşyk

```
“hello\n”
```

tipli literlerden ybarat setiriň soňunda hem saklanýar. Häzirki ýagdaýda soňunda '\0' liter ýerleşdirilen, literlerden düzülen massiw emele getirilýär.

h	e	l	l	o	\n	\0
---	---	---	---	---	----	----

printf – iň formatynda %s spesifikasiýa, oňa degişli argument ýokardaky ýaly görnüşdäki string(setir) bolmalydyr. copy functiony hem öz işleýişinde okaýan argumenti '\0' litera bilen gutarýar hasaplaýar, özünem beýlekiler bilen bir hatarda ol literany hem kopirleýär.(Hemme aýdylanlar adaty tekstiň içinde '\0' litera duş gelmeýär hasaplaýar.)

Hatda şular ýaly kiçijek programmada hem käbir kynçylyklaryň ýüze çykýanlygyny belläo geçmek gerek. Mysal üçin, eger ýol bererlikden uzyn setir gabat gelse main näme iş etmeli? getline functiony ähtibar işleýär: eger massiw

dolsa, ‘\0’ literasy gabat gelmedik halatynda hem geçirmekligi tamamlayar. Baş programma getline functiondan girizilen setiriň uzynlygyny alanyndan sön, onuň uzynlygy MAXLINE bilen gabat gelýän bolsa, onda bu aýratyn ýagdaýy gözöňünde tutmak üçin programmada käbir zatlary goşmak bolar. Programmanyň gysga bolmagynyň hatyrasyna biz oňa bu ýerde seretmeýäris.

getline functionynyň ulanyjylary girizilýän setiriň nä derejede uzyn boljaklygyny öňünden bilmeýärler, şonuň üçin getline çogmaklyk(overflow) şertini barlaýar. copy functionynyň ulanyjylary üçin kopirlenýän setiriň uzynlygy belli(ýa-da olar bu barada bilip bilerler), şonuň üçin hem bu ýerde goşmaça barlagyň geregi ýok.

Ýumuş 1. Ýokardaky programmanyň main-yny ol islendik uzynlykdaky iň uzyn setiri, onuň ölçegine hiç-bir çaklendirmesiz çapa çykarar ýaly edip täzeden ýazyň.

Ýumuş 2. Girizilýän setirleriň içinden uzynlygy boýunça 80 literadan uzynlaryny çapa çykarmak üçin programma düzüň.

Ýumuş 3. Hebir girizilýän setirde yzly-yzyna gelýän probel we tabulýasiýa literalaryny bir probele çalyşýan hem-de boş setireri aýyrýan programma düzüň.

Ýumuş 4. s stringdäki literalary ters tertipde ýerleşdirýän reverse(s) functionyny düzüň. Her bir girizilýän setiri tersine öwürýän programma düzüň, ol programmada bolsa reverse(s) functionyny ulanyň.

Daşky üýtgeýän ululyklar we olaryň täsir edýän çägi

line, longest üýtgeýän ululyklary diňe main functionyna degişli, ýa-da aýdylyşy ýaly şol fubctionyň içinde lokallaşdyrylan. Olar main-iň içinde kesgitlenen bolansoň, başga hiç bir function göniüden-göni olara ýüzlenip bilmeýär. Beýleki functionlaryň üýtgeýän ululyklary barada hem şony aýtmak bolar; mysal üçin getline-daky i-iň, copy-daky i bilen hiç-hili baglanyşygy ýokdur. Functionyň her bir lokal üýtgeýän ululygy, şol functiona ýüz tutulan pursatynda döreýär we functiondan çykylanda bolsa olar ýitip gidýärler. Olar ýaly üýtgeýän ululyklara, beýleki dilleriň terminologiýasynda aýdylyşy ýaly *awtomatik* üýtgeýän ululyklar diýilýär.(4-nji bapda, lokal ululyklara çagyryşlaryň arasynda hem öz bahasyny saklap bilmek mümkinçiligini döredýän static diýilip atlandyrylýan ýadyň klasyna seredilip geçilýär.)

Şeýlelikde awtomatiki ululyklar functiona girilmegi we ondan çykylmagy bilen birwagtda emele gelýär we ýitip gidýär, olar öz bahalaryny ol çakylykdan beýleki bir çakylyga çenli saklamaýarlar, şonuň üçin functiona täzeden ýüzlenilende olaryň bahalary täzeden görkezilmelidir. Eger şeýle edilmese onda olar özünde “musor” saklarlar.

Awtomatik ululyklaryň alternatiwasy görnüşinde, atlary boýunça islendik functiondan ýüzlenmeklige mümkinçilik berýän *daşky* ululyklary mysal getirmek bolar.(Bu mehanizm Fortrandaky ýadyň comman bölegine we Pascaldaky iň daşky blokda beýan edilen ululyklar bilen meňzeşdir.) Şeýlelikde, *daşky* ululyklara hemme functionlardan ýüzlenip bolýanlygy üçin, olary functionlary maglumat boýunça baglanyşdyrýan argumentleriň ornunda ulanmak bolar. Bulardan başgada daşky ululyklar, functionlar aktiwleşende ýüze çykman, hemişe barlygy üçin olar

function öz işini tamamlandan soň hem şol ýerde kabul eden bahalaryny özlerinde saklaýarlar.

Daşky ululyklar hemme functionlaryň daşynda, diňe bir gezek kesgitlenýärler; şonüň üçin olara özbaşdak ýat bölünip berilýär. Daşky ululyklar olary ulanmakçy bolýan hemme functionlarda beýan edilmelidir(deklerasiýa). Bu beýan edilmegi ululygyň tipi barada maglumat saklamalydyr. Beýan etmeklik açyk extern instruksiýasy görnüşde ýa-da gerek maglumat kontekstden alynýan açyk däl görnüşde bolup biler. Aýdylanlary has takyklaşdyrmak üçin, iň uzyn setiri çapa çykarýan programmany line, longest we max ululyklary daşky ululyk görnüşinde beýan etmek arkaly gaýtadan ýazalyň. Bu üç functionlaryň hemmesiniň çagyryşynda, beýan edilişinde we göwresinde üýtgeşmeleriň girizilmegini talap edýär.

```
#include <stdio.h>
#define MAXLINE 1000 /* girizilýän setiriň max. uzynlygy */
int max; /* seredilen setirleriň içinden iň uzyny */
char line[MAXLINE]; /* häzirki setir */
char longest[MAXLINE] /* iň uzyn setir */
int getline(void);
void copy(void);
/* n uzyn setiri çap etmek */
main()
{
    int len;
    extern int max;
    extern char longest[];
    max = 0;
    while((len = getline()) > 0)
        if (len > max) {
            max = len;
            copy();
        }
    if (max > 0) /* iň bolmanda bir setir */
        printf("%s", longest);
    return 0;
}

/* getline: ýörüteleşdirilrn görnüşi */
int getline(void)
{
    int c, i;
    extern char line[];
    for(i = 0; i < MAXLINE-1
        && (c = getchar()) != EOF && c != '\n'; ++i)
        line[i] = c;
```

```

if (c == '\n') {
    line[i] = c;
    ++i;
}
line[i] = '\0';
return i;
}

/* copy: ýörüteleşdirilen görnüşi */
void copy(void)
{
    int i;
    extern char line[], longest[];
    i = 0;
    while ((longest[i] = line[i]) != '\0')
        ++i;
}

```

main, getline we copy üçin daşky ululyklar biziň mysalymyzyň başynda kesgitlenýär, olaryň tipi kesgitlenýär we ýer bölünip berilýär. Daşky ululyklary kesgitlemek, lokal ululyklary kesgitlemekden tapawutlanmaýar, diňe olaryň functionlaryň çäginde daşarda ýatanlygy üçin olara daşky ululyklar hasaplanylýar. Functionyň daşky ululygy ulanyp bilmegi üçin, ilki bilen oňa ol ululyk barada habar bermeli. Bu bolsa *extern* sözüniň kömegi bilen beýan edilýär, bu beýan etmeklik daşky ululygy beýan edilişinden diňe *extern* söziden başlanýanlygy bilen tapawutlanýar.

Käbir ýagdaýlarda *extern* beýan etmesini ýazman hem bolar. Eger programma ýazylan faýlda daşky ululygyň kesgitlenişi ony ulanýan functiondan ýokarda bolsa, onda *extern* beýan etmesiniň zerurlygy ýok. Şeýlelikde main, getline we copy functionlarynda *extern* beýan edilmesiniň geregi ýok. Adatça daşky beýan etmeklik faýlyň başynda ýerleşýär, şonuň üçin hem hemme functionlar üçin *extern* beýan etmeleriniň geregi bolmaýar.

Eger programma birnäçe faýlda ýerleşip, daşky ululyk file1-da kesgitlenip, file2-de we file3-de ulanylýan bolsa onda file2-de we file3-de *extern* beýan etmeklik hökmandyr, sebäbi hemme üç faýldaky functionlar şol bir daşky ululyga ýüzlenýärler. Praktikada hemme daşky ululyklaryň we functionlaryň beýan edilişini baş faýl diýilip atlandyrylýan (header-file) aýratyn faýlda ýazmak we programmanyň her bir faýlynyň başynda *#include* sözüniň kömegi bilen ony ýerleşdirmek amatly bolýar. Umumy şertleşik boýunça header file adynyň yzynda *.h* goýulýar. *<stdio.h>* header faýlda hem standart bibliotekanyň functionlary beýan edilýär. Header faýllar barada 4-nji bapda, standart biblioteka barada bolsa 7-nji bapda we B goşmaçada durulup geçiler.

Görnüşi ýaly getline we copy functionlarynyň ýörüteleşdirilen görnüşleriniň argumentleri ýok, şonuň üçin hem olaryň prototiplerini hem getline() we copy() ýaly ýazmak amatly ýaly bolup görünýär. Taze standart köne C programmalar

bilen ylalaşygy saklar ýaly, argumentleriň bolmazlygyny deňişli argumentleriň gabat gelyänligini barlamaly dăldigi barada signal hökmünde kabul edýär. Şonuň üçin hem haçanda barlagy saklamaly we argumentleriň ýoklugyny aýyk görkezmeke isleseň void sözünden peýdalanmanmaly. Biz bu meselä 4-nji bapda gaýdyp geleris.

Biz bu bölümde daşky ululyklar barada *kesgitlemek* hem-de *beýan etmek* düşüňjelerini örän seresaplylyk bilen ulanandygymyzy belläp geçmek gerek. “Kesgitlemek”-de ululyk döredilýär we oňa maşynyň huşunda ýer bölünip berilýär. “Beýan” etmekde ululygyň tebigaty fiksirlenýär, ýöne onuň üçin hiç-hili ýer bölünip berilmeýär.

Hemme ululyklary daşky ululyk görnüşinde beýan etmeli diýen pikiriň hem bardygyny belläp geçmek gerek. Ilkinji gezek göräýmäge baglanyşyk ýönekeýleşýän ýaly bolup görünýär – argumentleriň toplumy gysga, ululyklara gerek bolan islendik ýerden ýüzlenip bolýar; ýöne ol ululyklar gerek däl ýerinde hem elýeterli bolup galýar. Daşarky ululykary juda köp ulanmak maslahat berilmeýär, sebäbi maglumatlar boýunça baglylyk düşnüksiz bolup galýar, üýtgeýän ululyklar garaşylman durka we gizligin usullar bilen üýtgäp bilmegi mümkin. Bulardan başgada beýle programmalary kämilleşdirmek kyn bolýar. Şol sebäplere görä, iň uzyn setiri tapýan programmanyň ikinji görnüşü birinji görnüşinden ýaramazdyr, olardan başgada iki peýdaly functionlaryň içinde hasaplamak üçin takyk ululyklaryň ulanylmagy bilen umumylyklarynyň bozulmagy,

Biz C diliniň ýadrosy diýilip aýdyp bolaýjak zatlara seredip geçdik. Beýan edilen “kerpiçjikler” uly ölçegli peýdaly programmalary düzmek üçin ýeterlikdir, şonuň üçin hem siz biraz wagtlap okamaklygy bes edip, dürli programmalary düzseňiz örän oňat bolar. Biz aşakdaky ýumüşlarda, öňki ýumüşlara garanynda birnäçe esse kyn bolan meseleleri hödürleýäris.

Ýumuş 1. Girizilýän tekstde tabulýasiýa literalaryny, gerek bolan boşluklar bilen çalşyryan detlab programmasyny düzüň. Her bir tabulýasiýa islendik ýerden başlap kesgitli n sany orny eýeleýär. Bu ýerde n -i üýtgeýän ululygyň bahasy görnüşinde bermek amatlymy ýa-da atlandyrylan hemişelik görnüşde?

Ýumuş 2. Çapa çykarylýan tekstiň görnüşü üýtgemez ýaly, boşluklaryň yzygiderligini iň az mukdardaky tabylýasiýa literleri we boşluklar bilen çalyşýan enlab programmasyny düzüň. Edil detlab programmasyndaky ýaly bir tabulýasiýa literasynyň näçe boşluk bilen çalşyrylmalydygyna üns beriň.??

Ýumuş 3. Giriş akymyndaky literalary n -nji orundan saga geçmez ýaly edip çapa çykarmak üçin programma düzmeli. Munuň özi, uzynlygy n -den geçýän setirlerleriň soňky bölegi indiki setire geçirilip çapa çykarylmaladygyny aňladýar. Setiriň indiki setire geçirmeli bölegini n -nji orundan çepde, iň soňky boşluk däl literadan soň “gözlemel”. Düzen programmaňyzyň örän uzyn setirler bilen, şeýle-de n -nji orna çenli hiç-hili boşluk, tabulýasiýa bolmadyk setirler bilen hem dogry işlemegi üçin alada ediň.

Ýumuş 4. Islendik C programmadan hemme düşüdirişleri (kommentariýalary) aýyryan programma ýazyň. Literleriň stringini we string hemişeliklerini bilen işläniňizde ünsli boluň. C dilinde düşündirişler iç-içine goýulmaýar.

Ýumuş 5. C programmadaky skobkalaryň hemme görnüşiniň balansyny (jübütligini) barlamak bilen ýönekeýje sintaksis barlag geçirýän programma ýazmaly. Dyrnaklar (goşa hem-de ýeke), eskeýp yzygiderlikler (\...) we kommentariýalar barada hem ýatdan çykarmaň. (Hemme ýagdaýlary gözöňünde tutup ýazjak bolsak, onda - bu çylşyrymly programmadar)

Tipler, operatorlar we aňlatmalar

Programmanyň iş salyşýan esasy obýektleri üýtgeýän ululyklar we hemişeliklerdir. Üýtgeýän ululyklar beýan edilişde sanalyp geçilýär, olaryň tipleri kesgitleňýär. Beýan edilişde üýtgeýän ululyklaryň başlangyç bahalary hem görkezilip biliner. Operasiýalar olaryň üstünden dürli täsirleri geçirýärler. Täze bahasyny almak üçin, aňlatma üýtgeýän ululyklaryň hem-de hemişelikleriň üstünde dürli operasiýalary geçirip biler. Obýektiň tipini onuň kabul edip biljek bahalarynyň we olaryň üstünden geçirilip bilinjek operasiýalaryň köplügi kesgitleýär. Bu babda şol ady agzalan “kerpiçjagazlar” barada gürrüň ederis.

ANSI standarty tarapyndan esasy tiplere we aňlatmalara ujypsyz üýtgetmeleriň we goşmaçalaryň köplügi tassyklandy. Indi islendik bitin tip alamatly (**signed**) we alamatsyz, (**unsigned**) bolup biler. Alamatsyz konstantalaryň we onaltylyk liter konstantalarynyň ýazgylary gözöňünde tutuldy. Ýüzýän nokatlar bilen geçirilýän operasiýalar indi birleýin takyklykda hem bolup biler. Ýokarlandyrylan takyklygy üpjün edýän **long double** tipi girizildi. Indi string ululyklar kompilýasiýa wagtynda konkatenirlenýär (“birleşdirilýär”). Tipiň belli bir aralygynda gurnalýan sanalýan tipler C diliniň bir bölegi bolup galdy. Bolup biläýjek tötänleýin üýtgetmelerden goramak maksady bilen obýektleri `conts` sözi bilen bellemeklige rugsat berilýär. Täze tipleriň girizilmegi bilen baglanyşyklykda, bir arifmetiki tipden beýlekisine awtomatiki öwürmekligi düzgüni giňeldildi.

Üýtgeýän ululyklaryň atlary

Biz 1-nji babda bu barada hiç zat aýtmadyk hem bolsak, üýtgeýän ululyklaryň we atlandyrylan hemişelikleriň atlarynda käbir çaklendirmeler bar. Atlar harplardan hem-de sanlardan ybarat bolup bilýärler, özünem birinji litera hökman harp bolmalydyr. Aşaky çyzyk belligi “_” harp hasaplanylýar; ony käbir uzyn atlar ýazylanda okamaklyk düşnükli bolar ýaly edip ulanmaklyk amatly bolýar. Üýtgeýän ululyklaryň atlaryny “_” literasy bilen başlamak maslahat berilmeýär, sebäbi biblioteka programmalarynyň köpüsy hut şol litera bilen başlanýar. Uly (baş) we kiçi (setir) harplar biri-birinden tapawutlanýarlar, ýagny `x` we `X` dürli atlardyr. Adat bolşy ýaly C programmada kiçi harplar bilen üýtgeýän ululyklar, uly harplar bilen bolsa atlandyrylan hemişelikler belleniýär.

Içki atlar üçin ilkinji 31 literanyň ahmiýeti bardyr. Functionlaryň we daşky ululyklaryň atlary üçin bolsa 31 literadan az hem bolup biler, sebäbi ol atlar

assembler we ýükleýjiler tarapyndan ulanylýar, dil tarapyndan olar barlanylmaýar. Daşky atlaryň unikallygyna(?) haýsy registerdeligine seretmezden diňe ilkinji 6 literiň çäginde güwä geçilýär. Diliň özünde ulanylýan sözler bolan if, else, int, float, we ş. m. eldegrilmesiz bolup, olar üýtgeýän ululyklaryň atlary hökmünde ulanmak bolmaýar. Olaryň hemmesi kiçi registerde ýazylýar(ýagny kiçi harpda).

Elbetde üýtgeýän ululyklara olaryň niýetlenişi bilen baglylykda biri-birinden tapawutly manyly atlaryň berilmegini maslahat berýäris. Biz lokal ululyklar üçin, esasanam gaýtalanýan gurluşlaryň sanaýjylary üçin gysgajyk atlaryň, has uzyn atlary bolsa daşky ululyklar üçin ulanmaklygy amatly hasaplaýarys.

Maglumatlaryň tipleri we ölçegleri

C dilinde bary ýogy birnäçe baza tipa bar:

char	bir baýt bolup, ol ýol berilýän literalardan diňe birini saklap bilýär.
int	bitin san bolup ol ulanylýan maşyndaky bitin sanlaryň tebigy ölçegini görkezýär (?).
float	ýüzýän nokatly, birleýin takykly san.
double	ýüzýän nokatly, ikileýin takyklykdaky san.

Bulardan başgada birnäçe kwalifikatorlar bolup, olary görkezilen baza tipleri bilen birlikde ulanmak bolýar. Mysal üçin, short (gysga) we long (uzyn) kwalifikatorlary bitin tipler üçin ulanylýar:

short int sh;

long int counter;

Beýle görnüşli beýan etmelerde adatça edilişi ýaly int sözünü görkezmän hem bolar.

Bitin short we bitin long dürli uzynlykly bolmaly, a int bolsa ulanylýan maşyndaky bitin sanyň tebigy ölçegine gabat gelýär. Köplenç ýagdaýda short kwalifikatory bilen ýazylan bitin sanlar üçin 16 bit , long kwalifikatorly bitin sanlar üçin bolsa 32 bit ýer berilýär, int tipli ululyklar üçin bolsa 16 ýa-da 32 bit berilýär. Kompilýatory döredijiler ulanylýan kompýuterleriň karakteristikasyňa baglylykda, aşakdaky çäklendirmeleri gözönünde tutmak bilen bitin tipleriň ölçeglerini özleri saýlap bilerler: short we int tipleriň ölçegleri azyndan 16 bit, long tipiniňki bolsa 32 bit, short-yň ölçegi int-iň ölçeginden uly bolmaly däl, int-iň ölçegi bolsa long-yň ölçeginden uly bolmaly däl.

signed (alamatly) ýa-da unsigned (alamatsyz) kwalifikatorlaryny char ýa-da islendik bitin tipler üçin ulanmak bolar. unsigned tipine degişli ululygyň bahasy elmydama položitel ýa-da nula deňdir we moduly boýunça 2^n arifmetiki kanyna boýun egýär, bu ýerde n tipiň ikilik ýazgysyndaky bitleriň mukdary. Mysal üçin char baha üçin 8 bit berilýän bolsa, onda unsigned char -128... +127 bahalary kabul eder (maşynda ikilik doldurgyç kodlary). Ýöne char tipiniň alamatly ýa-da alamatsyzlygy maşyna bagly, ýöne islendik ýagdaýda hem çap edilýän literleriň kodlary položiteldir.

long double ýokarlandyrylan takyklykdaky ýüzýän nokatly arifmetika üçin niýetlenendir. Bitin sanlarda bolşy ýaly, ýüzýän nokatly obýektleriň ölçegleri hem

kompilýatora baglydyr; float, double we long double şol bir ölçegli ýa-da iki ýa-da üç dürli ölçegli hem bolup bilerler.

Islendik ölçegli atlandyrylan hemişelikler maşyniň we kompilýatoryň beýleki karakteristikalary bilen birlikde standart baş <limits.h> we <float.h> faýllarda saklanýar. (B goşmaça serediň)

Ýumuş 1. Standart baş faýllardan degişli bahalary çap etmek we hasaplamak arkaly signed şeýle hem unsigned görnüşde ýazylan char, short, int we long tipleriň kabul edip bilýän bahalarynyň çäklerini berýän programma düzmeli. Dürli tipli ýüzýän nokatly sanlaryň çäklerini kesgitleň. Bu çägi kesgitlemek bitin sanlaryňka garanyňda çylşyrymlydyr.

Hemişelikler

Bitin hemişelik, mysal üçin 1234 int tipine degişlidir. long tipli konstanta l ýa-da L bilen tamamlanýar, nysal üçin 123456789L; int tipi bilen görkezip bolmaýan juda uly bitin sanlar, long tipi arkaly beýan edilýärler. Alamatsyz hemişelikler u ýa-da U literalar bilen tamamlanýarlar, eger ul ýa-da UL bilen tamamlanýan bolsa onda hemişeligiň tipi - unsigned long.

Ýüzýän nokatly hemişelikleriň onluk nokady (123.4) ýa-da eksponensial bölegi (1e-2) bolýar ýa-da olaryň ikisi hem bolýar. Eger olaryň soňlamasy (окончания) bolmasa onda olar double tipine degişli hasaplanylýar. f we F soňlamalar float tipine degişlilik görkezýär, a l we L soňlamalar bolsa long double tipine degişlilik görkezýär.

Bitin onluk sanlardan başgada bitin sekizlik we bitin onaltlyk sanlar hem bolup bilerler. Eger hemişelik nuldandan başlanýan bolsa, onda ol sekizlik görnüşde ýazylypdyr, egerde hemişelik 0x-den ýa-da 0X başlanýan, bolsa onda ol onaltlykda ýazylypdyr diýilip hasaplanylýar. Mysal üçin onluk bitin san bolan 31-i 037-i ýa-da 0X1F görnüşde ýazmak bolar. Sekizlik we onaltlyk hemişelikleriň ýazgysy L soňlama (bu ýerde L long tipi görkezýär) ýa-da U (hemişelik alamatsyz, unsigned) bilen tamamlanyp biler. Mysal üçin 0XFUL 15-e deň bolup unsigned long tipe degişli bolar.

Liter hemişelikler ‘x’ görnüşdäki ýaly birleýin dynnagynyň içinde ýazylan literadan ybarat bitin ululykdyr. Onuň san bahasy, berlen maşyndaky literleriň ýygindysyndaky literiň kodyna deňdir. Mysal üçin ASCII kodirowkada ‘0’ hemişelik 48-e deň, görnüşi ýaly onuň 0 san bilen hiç-hili dahyly ýokdyr. Eger biz haýsydyr bir baha (mysal üçin 48) ýazman ‘0’ ýazsak, onuň san bahasy kodirleniş usulyna laýyklykda kesgitlener, ýöne ol ýazgy liter ululyk görnüşinde kodirmek usulyna bagly däl we adaty sandan has düşnükli. Liter hemişelikler, köplenç ýagdaýda beýleki literler bilen deňeşdirmek maksady bilen ulanylýan hem bolsa, olar islendik beýleki bitin tipler ýaly, sanlaryň üstünden geçirilýän operasiýalarda hem ulanylyp biliner.

Birnäçe literler, liter we setir hemişeliklerinde Esc yzygiderligiň kömegi bilen ýazylyýarlar, mysal üçin \n (täze setir); beýle yzygiderlikler iki sany literiň kömegi bilen ýazylýan hem bolsalar, olar bir literi aňladýar. Mundan başgada islendik sekizlik kody aşakdaky ýaly ýazmak bolar

‘\000’

bu ýerde 000 – bir, iki ýa-da üç sekizlik san(0...7) ýa-da
‘\xhh’

bu ýerde hh – bir , iki ýa-da köp onaltlyk sanlar(0...9, a...f, A...F). Şeýlelikde biz aşakdaky görnüşde ýazyp bileris

```
#define VTAB ‘\013’          /* ASCII-däki wertikal tabulýasiýa */
#define BELL ‘\007’          /* ASCII-de jaň */
```

ýa-da onaltlyk görnüşde:

```
#define VTAB ‘\xb’          /* ASCII-däki wertikal tabulýasiýa */
#define BELL ‘\x7’          /* ASCII-de jaň */
```

Esc yzygiderlikleriň doly toplumu aşakdakylardan ybaratdyr:

\a	jaň-sinaly	\\	ters ýapgyt çyzyk
\b	bir ädim yza	\?	sorag belgisi
\f	sahypany geçirmek	\’	birleýin dyrnak
\n	täze setir	\”	goşa dyrnak
\r	karetkany yzyna gaýtarmak	\000	sekizlik kod
\t	kese tabulýasiýa	\xhh	onaltlyk kod
\v	dik tabulýasiýa		

‘\0’ litera belgisi – ol nul bahaly litera, ýa-da aýdylyşy ýaly null litera. Aňlatmanyň litera tebigatyny görkezmek üçin 0 -yň ýerine ‘\0’ ýazýarlar, aslynda ol ý-da beýleki ýagdaýda hem ýazgy nuly aňladýar.

Aňlatma hemişeligi – diňe konstantalaryň üstünden operasiýa geçirýän aňlatmadyr. Beýle aňlatmalaryň bahasy hasaplama wagtynda dälde kompilyasiýa döwründe hasaplanylýar, şonuň üçin hem olary hemişelikleriň ulanylýan islendik ýerinde ulanyp bolar, mysal üçin

```
#define MAXLINE 1000
char line[MAXLINE+1];
```

ýa-da

```
#define LEAP 1 /* in leap years – uzak ýyllar */
int days[31+28+LEAP+31+30+31+30+31+31+30+31+30+31];
```

Setir hemişeligi ýa-da setir literaly – goşa dyrnagyň içinde ýazylan nul ýada ondan köp bolan literalardyr, mysal üçin

“Bu setir hemuşeligi”

ýa-da

```
“” /* boş setir */
```

Goşa dyrnaklar setiriň içine girmeyär, olar diňe setiriň çäginde görkezýär. Liter hemişeliklerindäki ýaly, setir hemişelikleriniň düzümine Esc yzygiderlikleri hem goşmak bolar; mysal üçin \” goşa dyrnagy aňladýar. Setir hemişeliklerini kompilyasiýa döwründe konkatenirmek (birleşdirmek) bolýar; mysal üçin aşakdaky iki setiriň ýazgysy

“Salam” “Dünýe!”
aşakdaky bir setiriň ýazgysyna deňdir:

“Salam Dünýe!”

Setirleriň bu häsiýeti uzyn setirleri böleklere bölmäge we bu bölekleri aýratyn setirde saklamaga mümkinçilik berýär.

Setir hemişeligi – munuň özi literalaryň massiwidir. Setir maşynyň ýadynda ýerleşende onuň ahyrynda hökman ‘\0’ litera durýar, şol sebäpli setir üçin niýetlenen kompýuteriň ýady onuň öz uzynlygyndan bir baýt köp ýer tutýar. Bu döredilýän setiriň uzynlygyna çäk goýulmaýanlygyny aňladýar, ýöne onuň uzynlygyny kesgitemek üçin tutuş setire seredip geçmek zerurdyr. `strlen(s)` s setiriň iň soňky ‘\0’ literasyny göz önünde tutmazdan, onuň uzynlygyny hasaplaýar. Aşakda ol functionyň biziň tarapymyzdan ýazylan programmasyny getirýäris:

```
/* strlen: s setiriň uzynlygyny kesgitleýär */
int strlen(char s[])
{
    int i;
    i = 0;
    while (s[i] != '\0')
        ++i;
    return i;
}
```

`strlen` we setirler üçin niýetlenen beýleki functionlar `<string.h>` standart baş faýlda ýazylandyr.

Uns beriş we ýatda saklaň, bir literany saklaýan liter ululyk bilen bir literany saklaýan setir ululyk şol bir zat däl: ‘x’ bilen “x” şol bir zat däl. ‘x’ ýazgy x literanyň kodyna deň bolan bitin sandyr, “x” ýazgy bolsa literalaryň massiwi bolup ol häzirki wagtda bir literany (x harpy) we ‘\0’-y özünde saklaýar.

C dilinde hemişelikleriň ýene bir görnüşi, sanalýan hemişelikler ulanylýar. Sanalýan hemişelikler – bitin hemişelikleriň aşakdaky ýaly

```
enum boolean {NO, YES}
```

sanalýan görnüşidir. `enum` (sözün manysy - sanalmak)-daky birinji at 0 baha eýe, indili 1, we ş.m. (eger hemişelikleriň bahalary aýk görkezilmedik bolsa). Eger hemme hemişelikleriň bahalary görkezilmedik bolsa, onda aşakdaky iki mysaldaky ýaly, iň soňky görkezilen bahadan başlap artmaklygy (prograssiýa) dowam etdirýärler:

```
enum escapes {    BELL = '\a', BASCSPACE = '\b', TAB = '\t',
                  NEWLINE = '\n', VTAB = '\v', RETURN = '\r'};
```

```
enum months {    JAN = 1, FEB, MAR, APR, MAY, JUN,
                  JUL, AUG, SEP, OCT, NOV, DEC };
/* FEB 2-ä, MAR 3-e deň, we ş.m. */
```

Atlar dürli sanalmalarda biri-birinden tapawuplanmalydyrlar. Şol bir sanalmanyň içinde bahalar gabat gelip bilerler.

enum serişdesi hemişeliklere at dakmaklygyň amatly usulydyr, özünem #define-den tapawutlylykda hemişelikleriň bahalary awtomatik döredilip hem biliner. Sanalýan tipleri üýtgeýän ululyklary kesgitlemek üçin ulanmaklyga ygtyýar berilýär, ýöne ol ululyklara dakylýan baha olaryň tipine gabat gelýänligini barlamak kompilýatoryň wezipesine girmeyär. Ýöne diňe, şeýle mümkinçiligiň barlygynyň özi hem, enum-yň #define-den gowydygyny görkezýär. Bulardan başgada debug(отладчик) sanalýan ululyklaryň bahalaryny simwol görnüşde çap etmäge mümkinçilik alýar.

Deklarirlemek

Hemme ululyklar özleriniň ulanylmazyndan öň deklarirlenmelidirler, özünem käbir deklarasiýalar aýyk däl görnüşde berilip bilener. Deklarirlemek munuň özi tipi, hemde şol tipe degişli bolan üýtgeýän ululyklaryň birini ýa-da toplumyny görkezmekdir, mysal üçin:

```
int    lower, upper, step;  
char  c, line[1000];
```

Üýtgeýän ululyklar erkin görnüşde dürli hili deklarirlenip biliner, mysal üçin ýokardaky deklarirlenmekligi aşakdaky görnüşde hem ýazmak bolar:

```
int    lower;  
int    upper;  
int    step;  
char  c;  
char  line[1000];
```

Deklarirlemekligiň soňky görnüşi köp ýer alýan hem bolsa, ol gowydyr, sebäbi her bir deklarirlenen ululyga kommentariý (düşündiriş) ýazmaga mümkinçilik berýär we gelejekgi kämilleşdirmek üçin has amatlydyr.

Üýtgeýän ululyk deklarirlenýän döwründe onuň başlangyç bahasy görkezilip hem biliner, mysal üçin:

```
char  esc = '\\'  
int    i = 0;  
int    limit = MAXLINE+1;  
float  eps = 1.0e-5;
```

Awtomatik däl üýtgeýän ululyklaryň inisializasiýasy (başlangyç bahany dakmak) diňe bir gezek, programma ýerine ýetip başlamazyndan öň ýerine ýetirilýär, özünem onuň inisializatory hemişelik aňlatma bolmalydyr. Aýyk inisializirlenýän awtomatik üýtgeýän ululyklar her gezek functiona ýa-da bloga girilende başlangyç bahany alýar, onuň inisializatory islendik aňlatma bolup biler. Daşky we statik üýtgeýän ululyklaryň başlangyç bahasy görkezilmedik halatynda nula deň bolýar. Aýyk görnüşde inisializirlenmedik awtomatik üýtgeýän ululyklar kesgitsiz bahalary saklaýarlar (“musor”, “zibil”).

Deklarasiýada Islendik üýtgeýän ululyk üçin, onuň bahasynyň soňra üýtgemeyänligini görkezmek maksady bilen const kwalifikator ulanylyp biliner.

```
const double e = 2.71828182845905;  
const char msg[] = “duýduryş”;
```

const kwalifikatorly massiwler üçin ulanylanda, ol massiwiň hiç bir elementiniň üýtgedilip bilinmejekdigini aňladýar. const görkezmesi argument massiw üçin hem ulanylyp biliner, beýle ýagdaýda function ol massiwi üýtgedip bilmeýär:

```
int strlen(const char[]);
```

const kwalifikator bilen bellenen üýtgeýän ululygyň bahasynyň üýtgediljek bolundaky reaksiýa kompilyatorya baglydyr.

Arifmetiki operatorlar

+, -, *, / şeýle hem modul almak % operatory binar arifmetiki operasiýalardyr. Bitin sanlar özara bölünen ýagdaýynda olaryň drob bölekleri taşlanylýar.

```
x % y
```

aňlatma x ululyk y ululyga bölünendäki galyndyny kesgitleýär, eger-de ol tutuşlygyna bölünýän bolsa aňlatmanyň bahasy nula deň bolar. Mysal üçin eger ýyl 4-e galyndysyz bölünýän bolsa (ýöne 100- bölünmeýän bolsa) ol ýyl uzak ýyl hasaplanylýar. Mundan başgada 400-e bölünýän ýyllar hem uzak ýyllar hasaplanylýar. Şeýle ýazyp bileris

```
if ((year % 4 == 0 && year % 100 != 0 || year % 400 == 0)
```

```
    print("%d uzak ýyl\n",year);
```

```
else
```

```
    printf("%d adaty ýyl\n",year);
```

% operatory float we double tipli operandlar üçin ulanylmaýar. / ýerine ýetirilende drob böleginiň haýsy tarapa kesiljekdigini (artýan ýa-da kemelýän tarapyna) we % operatory otrisatel operandlar bilen ýerine ýetende netijäniň alamatynyň nähili boljakdygy bu nasyna bagly.

+ we - binar operatorlaryň birmeňzeş prioritetleri (ýerine ýetiş tertibi) bar. Olaryň prioritetleri *, / we % operatorlaryň prioritetlerinden aşakdayr, olaryň prioritetleri bolsa öz gezeginde + we - unar operatorlaryňkydan aşakdadyr. Şol bir prioritetli arifmetiki operasiýalar çepden başlap saga tarap ýerine ýetýärler.

Bu babyň soňunda ýerleşdirilen 2.1 tablisada hemme operatorlaryň prioritetleri we olaryň ýerine ýetiş tertipleri görkezilendir.

Gatnaşyk operatorlary we logiki operatorlar

Aşakdaky operatorlara gatnaşyk operatorlarydyr

```
< >= < <=
```

Olaryň hemmesiniň birmeňzeş prioritetleri bardyr.

```
== !=
```

operatorlarynyň bolsa ýokarkylar bilen deňşdireniňde bir basgançak pes prioriteti bardyr. Gatnaşyk operatorlary arifmetiki operatorlara garanyňda pes prioritetlidir, şonuň üçin $i < \text{lim}-1$ aňlatma, biziň garaşyşymyz ýaly $i < (\text{lim}-1)$ aňlatma ýaly ýerine ýeter.

&& we || logiki operatorlar has gyzyklydyr. Aralarynda &&, || operatorlar duran aňlatma çepden saga hasaplanylýar we haçanda netijäniň çynlygy ýa-da ýalandygy belli bolanda hasaplalamaklyk togtadylýar. Köp C programmalar bu häsiýeti ulanýarlar, mysal üçin 1-nji bapdaky getline functionyndaky gaýtalanmak:

```
for (i=0; i<lim-1 && (c=getlchar()) != '\n' && c != EOF; ++i)
    s[i] = c;
```

Indiki literany okamazdan öň, onuň üçin s massiwde ýer bardygyny ýa-da ýoklygyny barlamaly, başgaça aýdanymyzda $i < \text{lim}-1$ şerti ýerine barlamaly. Eger bu şert ýerine ýetmýän bolsa, onda biz hasaplamany dowam etmeli däldiris, ýagny indiki literany okamaly däldiris. Şeýle hem getchar functiona ýüzlenmezden öň c ululygy EOF bilen deňeşdirmek nädogry bolardy; ýagny getchar functionyny çagyrmak we eýe bolmak ol barlagdan öň ýerine ýetirilmelidir.

&& operatorynyň möhümligi (prioriteti) || operatorynyňkydan ýokarydyr, ýöne olaryň möhümligi gatnaşyk we deňlik operatorlarynyňkydan pesdir; şonuň üçin hem

```
i<lim-1 && (c = getchar()) != '\n' && c != EOF
```

aňlatmada goşmaça skobkalar hokman däldir. Ýöne != operatorynyň möhümligi eýe bolmak operatorynyňkydan ýokarydyr, şonuň görä

```
(c = getchar()) != '\n'
```

aňlatmada ilki bilen eýe bolmaklygyň, soňra bolsa '\n' bilen deňeşdirmekligiň ýerine ýetmegi üçin skobka zerurdyr. Kesgitleme boýunça gatnaşyk ýa-da logiki aňlatmalaryň san bahasy eger olar çyn bolanda 1-e deň we yalan bolanda 0-a deň.

! unar operatory nul däl operandy 1-a we nul operandy bolsa 1-e öwürýär. Adatça ! operator

```
if (!valid)
```

ýaly konstruksiýalarda ulanylyp, ol

```
if (valid == 0)
```

ýazga ekwiwalentdir.

Haýsy görnüşdäki ýazgynyň gowydygyny aýtmak kyn. !valid görnüşdäki konstruksiýa ýeňil okalýar (“eger valid däl bolsa”), ýöne has çylşyrymly aňlatmalarda oňa düşünmekligiň kynyrak bolmagy hem mümkin.

Tipleri özgertmek

Eger operatoryň operandlary dürli tiplere degişli bolsa, onda olar käbir umumy tipe getirilýärler. Bu birnäçe düzgünleriň esasynda ýerine ýetirilýär. Adatça diňe hiç-hili maglumat ýitmezden kiçi diapazondaky bahaly operandlar, uly diapazondaky bahaly operandlara awtomatik görnüşde öwrülýärler, mysal üçin $f + i$ görnüşdäki aňlatmalarda bitin sanyň ýüzýän nokatly sana öwürülmegi. Indeks görnüşde ýüzýän nokatly sanyň ulanylmagy ýaly manysyz aňlatmalara ýol berilmeýär. Maglumatlaryň ýitip biljek aňlatmalary (aýdaly uzyn bitin üýtgeýän ululyklary gysga tiplere dakylanda ýa-da ýüzýän nokatly ululyklaryň bahasyny bitin tiplere dakylanda) ýazylsa, bu barada duýduryjy habarlar berlip biliner, ýöne olar ýol bererlikdir.

char tipiniň bahasy – bu umuman kiçi bitin ululykdyr we olary arifmetiki aňlatmalarda erkin ulanmak bolar, bu bolsa literleriň üstünden geçirilýän mümkin bolan manipulýasiýalary juda ýeňilleşdirýär. Mysal hökmünde sanlaryň yzygiderligini olaryň san ekwiwalentine öwürýän atoi functionyň ýönekeýje görnüşde ýazylyşyna seredip geçeliň.

```
/* atoi: s-i bitin sana öwürmek */
```

```

int atoi(char s[])
{
    int i, n;
    n = 0;
    for (i = 0; s[i] >= '0' && s[i] <= '9'; ++i)
        n = 10 * n + (s[i] - '0');
    return n;
}

```

Biziň 1-nji bapda aýdyşymyz ýaly

`s[i] - '0'`

aňlatma `s[i]`-da saklanýän literanyň san bahasyny berýär, ýagny '0', '1' we ş.m. üznüksiz osýän yzygiderligi emele getirýär.

`char` tipi `int` tipe öwürmek lower functiony bilen baglanyşykly, ol ASCII kodlar ýygynyndysyndan bir literany alýar we eger ol baş harp bolsa ony setir harpa öwürýär. Eger litera baş harp bolmasa onda lower ony üýtgetmeýär.

/* lower: c-ni setir harpa öwürmek; diňe ASCII üçin */

```

int lower(int c)
{
    if (c >= 'A' && c <= 'Z')
        return c + 'a' - 'A';
    else
        return c;
}

```

ASCII kodlar ýygynyndysynda bu programma dogry işlär, sebäbi şol bir baş we setir harplaryň arasyndaky uzaklyk meňzeş (eger olara san hökmünde seretsek) we mundan başgada latyn harplary dykyz ýerleşen, ýagny bu kodlar ýygynyndysynda A hem-de Z harplaryň arasynda diňe harplar ýerleşýär. EBCDIC kodlar ýygynyndysynda ýokarky şert ýerine ýetmeýär, şonuň üçin hem biziň programmamyz bu ýagdaýda harplardan beýleki literalary hem üýtgeder.

B goşmaçada beýan edilen standart baş faýl <ctype.h> haýsy kodlar ýygynyndysyndadygyna bagly bolmazdan literalary barlaýan we özgertýän functionlaryň maşgalasyny özünde saklaýar. Mysal üçin `tolower©` function, eger `c` uly registrde ýazylan bolsa ony kiçi registre öwürýär, şonuň üçin hem `tolower` functiony ýokarda seredilen lower functionyny islendik ýagdaýda çalyşýar.

`c >= '0' && c <= '9'`

barlagy

`isdigit©`

bilen çalşyrmak mümkin.

Indiden beýläk biz <ctype.h> -yň functionlaryny ulanarys.

Literi bitin sana öwürmekligiň bir kynçylygy bar: C dili `char` tipe degişli ululygyň alamatlydygyny ýa-da alamatly dälidigini kesgitlemeýär. `char`-y `int`-e öwrenimizde haçandyr bir wagt otrisatel bitin san alnyp bilermi? Dürli arhitekturaly maşynlar üçin dürli hili jogaplaryň bolmagy mümkin. Käbir maşynlarda uly biti birlik bolan `char` tipli ululyklar otrisatel bitin sanlara öwürülýärler (“alamatlarynyň köpelmegi” serişdesi boýunça). Beýleki

özürtmelerde bolsa char-y int-e öwürmek üçin onuň ikilik bitleriniň çepinden bir nuly goşulýar we elmydama položitel san alynýar.

Çap edilýän literleriň ýygynyndasyndaky islendik litera otrisatel bolup bilmejekligine garantiýa berilýär, şonuň üçin aňlatmalarda şeýle literalar elmydama položitel operandlaryň bolup gelýärler. Ýöne sekizbitlik (bitli islendik bir kod) char tipli üýtgeýän ululyk bir maşynda otrisatel san, beýleki bir maşynda bolsa položitel san bolup biler. Ylalaşyklygyň saklanmagy üçin liter däl ululyklary hem saklap bilýän char ululyklary beýan edilende signed ýa-da unsigned kwalifikatirlary ulanmak maslahat berilýär.

&& we || operatorlar bilen bilelikde ulanylýan $i > j$ tipli gatnaşyk we logiki aňlatma çyn bolsa 1 we ýalan bolsa 0 aňlatma-şerti kesgitleýärler. Şeýlelikde

$d = c >= '0' \&\& c <= '9'$

eýe bolmakda c ululyk san bolsa onda d ululygyň bahasy 1 bolar we c ululyk san bolmadyk ýagdaýynda 0 bolar. Ýöne netije çyn bolan ýagdaýynda isdigit ýaly functionlar noldan tapawutlanýan islendik bahany berip biler. if, while, for we ş.m.-de barlag geçirilende, şert nula deň däl bolsa ol çyn hasaplanylýar.

Açyk däl arifmetiki özürtme düzgün bolşy ýaly tebigy şekilde amala aşyrylýar. Umumy ýagdaýda iki operandly (binar operatorlar) + ýa-da * tipli operatorlar dürli alamatly operandly bolsalar, onda operasiýa başlamazdan ön kiçi tipler “öşýärler” we “uly” tiplere öwrülýärler. Öwürmek düzgünü takyk kesgitlenendir. Eger aňlatmada alamatsyz operandlar ýok bolsa onda aşakdaky formal däl düzgünler bilen çäklenmek bolar:

- Eger haýsy hem bolsa bir operand long double tipe degişli bolsa onda beýleki hem long double tipe getirilýär.
- Ýokardaky şert ýerine ýetmedik ýagdaýynda, operandlaryň biri double tipe degişli bolsa onda beýleki operand hem double tipe getirilýär.
- Ýokardakylar ýerine ýetmedik ýagdaýynda, operandlaryň biri float tipe geşişli bolsa onda beýleki operand hem float tipe getirilýär.
- Ýokardaky şertler ýerine ýetmedik ýagdaýynda char we short tipli operandlar int tipe getirilýär.
- we iň soňunda operandlaryň biri long tipli bolsa onda beýleki hem long tipe getirilýär.

float tipli operandlaryň awtomatik suratda double tipli operandlara getirilmeýänligine üns bereliň; şonuň bilen hem C diliniň bu wersiýasy ilkibaşdakysyndan tapawutlanýar. Umuman aýdanymyzda <math.h> bibliotekasynda ýygnaýan matematiki functionlar hasaplamalarda ikeldilen takyklykda işlemek üçin niýetlenendir. float esasan uly massiwlerde maşynyň ýadyny tygşytlamak maksady bilen we kähallatlarda ikileýin takyklykdaky arifmetika gymmat düşýän maşynlarda hasaplamany çaltlaşdyrmak üçin ulanylýar.

unsigned operandlaryň ýüze çykmagy bilen öwürmek düzgünü çylşyrymlaşýar. Esasy kynçylyk alamatly we alamatsyz bahalaryň deňeşdirilmegi bitin tipiň ölçegine bagly, bitin tipiň ölçeginiň bolsa dürli maşynlarda dürli hili bolmagy mümkin. Goý int tipiň bahasy 16 bit, a long tipiniň bahasy bolsa 32 bit

ýer tutýar diýeliň. Onda $-1L < 1U$, sebäbi $1U$ int tipine degişli we signed long tipine özgerdilýär we uly položitel san hökmünde kabul edilýär.

Özgerdilmek eýe bolmakda hem bolup geçýär: Eýe bolmaklygyň sag böleginiň tipi çep böleginiň tipine getirilýär, ol bolsa netijäniň tipi bolýar.

Litera alamatlarynyň köpeldilmegi arkaly ýa-da ýokarda beýan edilen beýleki usullaryň biri arkaly bitin sana öwrülýär.

Uly razrýadlarynyň taşlanmagy bilen uzyn bitin sanlar kiçi bitin sanlara ýa-da char tipe öwrülýär. Şeýlelikde aşakdaky programma böleginde

```
int i;  
char c;  
i = c;  
c = i;
```

c ululygyň bahasy üýtgemez. char tip int tipe öwrülende, alamatlaryň köpeldilmegine bagly bolmazdan elmydama dogrydyr. Ýöne eýe bolmaklygyň tertibini üýtgetsek onda maglumatyň ýitmegi mümkin.

Eger x float tipe degişli bolsa, a i int tipe degişli bolsa, onda $x = i$, $i = x$ instruksiýalar tipleriň özgerdilmegine getirer, özünem float tipi int tipe drop böleginiň taşlanmagy bilen amala aşyrylýär. Eger double tipi float tipine öwrülýän bolsa onda ýaha tegeleklenýär, ýa-da kesilýär; bu işiň ýerine ýetiriliş usulyna (realizasiýasyna) bagly.

Ýagny function çagyrylandaky argument aňlatmadyr, ol functiona geçirilende hem tipleriň özgerdilmesiniň bolmagy mümkin. functionyň prototipiniň bolmadyk ýagdaýynda char we short tipli argumentler int tipine öwrülýärler, float tipi bolsa double tipine öwrülýärler. Şonuň üçin biz haçanda functionlary çagyranymyzda char ýa-da float tipli argumentleri ulanan hem bolsak, ol argumentleri int ýa-da float görnüşde beýan etdik.

Islendik aňlatma üçin unar operatory ulanmak bilen onuň tipiniň özgerdilmegini aýyk görkezmek bolar. Aşakdaky ýaly

(tipiň-ady) aňlatma

ýazgy ýokarda sanalyp geçilen düzgünler boýunça aňlatmany skobkanyň içindäki görkezilen tipe getirýär. Getirmek operasiýasynyň manysyny şeýleräk gözöňüne getirmek bolar: aňlatma görkezilen tipli käbir üýtgeýän ululyga eýe bolýar we bu üýtgeýän ululyk bütün konstruksiýanyň ýerine ulanylýar. Mysal üçin bibliotekadaky sqrt programma double tipli argumentler üçin niýetlenendir we onuň argumentinde başga tipli argument görkezilse ol function manysyz zatlary yzyna gaýtaryp berer. (sqrt <math.h>-da beýan edilendir.) Şonuň üçin hem eger n bitin ululyk bolsa, onda biz aşakdaky ýaly ýazyp bileris

```
sqrt((double) n)
```

we n-iň bahasy functiona berilmezinden öň onuň tipi double tipine öwürüler. Bu ýerde n-iň bahasy double tipinde döredilýär, ýöne n-iň asul bahasy üýtgemeýär. Getirmek operatorynyň möhümligi (prioriteti) bu baby soňunda ýerleşdirilen tablisadan görnüşi ýaly islendik unar operatorlaryňkydan ýokarydyr.

Haçanda argumentler functionyň prototipinde beýan edilen bolsa, garaşylyşy ýaly, functionlar çagyrylanda gerekli tip özgertmeleri awtomatik bolup geçýär. sqrt functionyň aşakdaky ýaly


```
double sqrt(double);
prototipi bolan ýagdaýynda
    root2 = sqrt(2);
eýe bolmaklykda sqrt-ä ýüzlenilmezinden öň bitin 2 san açyk geçirmek
operasiýasyny görkezmezden awtomatik suratda double tipine (2.0) öwürler.
Geçirmek operasiýasyny, standart biblioteka girýän tötänleýin san generatorynyň
erkin wersiýasynda we functionda göreliň.
```

```
unsigned long int next = 1;
/* rand: tötänleýin 0..32767 aralykdaky bitin sanlary kesgitleýär */
int rand(void)
{
    next = next * 1103515245 + 12345;
    reurn (unsigned int) (next/65536) % 32768;
}

/* srand: rand üçin “zatravka”-ny goýýar */
void srand(unsigned int seed)
{
    next = seed;
}
```

Ýumuş 1. 0x ýa-da 0X bilen başlanýan onaltlyk san yzygiderligini degişli bitin sana öwürýän htoi(s) functionyny ýazyň. 0..9, a...f, a...F literlere onaltlyk sanlar diýilýär.

Inkrement we dekrement operatorlar

C dilinde üýtgeýän ululyklaryň bahasyny artdyrmak we kemeltmek üçin niýetlenen iki sany adaty bolmadyk operator bar. Inkrement operator ++ öz operandyna 1-i goşýar, a dekrement operator -- bolsa 1-i aýyrýar. Biz eýýäm birnäçe gezek üýtgeýän ululyklaryň bahasyny artdyrmak üçin ++ operatorndan peýdalandyk, mysal üçin

```
if (c == '\n')
    ++nl;
```

++ we -- operatorlara adaty dældigi, olary prefiks operatorlary (üýtgeýän ululygyň öňünden ýeleşdirmek ++n) we postfiks operatorlary (üýtgeýän ululygyň soňunda ýerleşdirmek n++) görnüşlerde ulanmak bolýanlygy üçindir. Ýöne ++n n-iň bahasynyň ulanylmazyndan öň onuň bahasyny ulaldýar, a n++ bolsa onuň bahasy ulanylandan soň onuň bahasyny ulaldýar. Goý n 5-i saklaýan bolsun, onda

```
x = n++;
```

x ululyga 5-i bäşi bahany dakar,

```
x = ++n;
```

x ululygyň bahasy 6-y bolar. Ol ýa-da beýleki ýagdaýda hem x-iň bahasy 6-a deň bolar. Inkrement we dekrement operatorlary diňe üýtgeýän ululyklar üçin ulanmak bolar. Mysal üçin, (i+j)++ ýazgy nädogrydyr.

```
if (c == '\n')
    nl++;
```

ýaly, diňe üýtgeýän ululygyň bahasyny kiçeltmek ýa-da ulaltmak üçin prefiks ýa-da postfiks operatorynyň haýsysyny ulananymyzyň tapawudy ýokdyr. Ýöne käbir ýagdaýlarda bu operatorlaryň diňe birini ulanmaly bolýar. Mysal üçin s stringden c bilen gabat gelýän hemme literalary aýyrýan squeeze(s,c) functionyna seredip geçeliň:

```
/* squeeze: s-den hemme c-leri aýyrýar */
void squeeze(char s[], int c)
{
    int i,j;
    for (i=j=0; s[i] != '\0'; i++)
        if (s[i] != c)
            s[j++] = s[i];
    s[i] = '\0';
}
```

Haçanda c-den tapawutlanýan litera gabat gelende ol j-nji oruna kopirlenýär we diňe şondan soň j-iň bahasy 1 san galdyrylýar, ýagny indiki literanyň kabul edilmegine taýýarlanylýar. Bu aşakdakylar bilen takyk gabat gelýär:

```
if (c == '\n') {
    s[i] = c;
    ++i;
}
```

has tygşyly hem ýazmak bolar:

```
if (c == '\n')
    s[i++] = c;
```

Üçünji mysal hökmünde t stringi s stringiň ahyrynda ýerleşdirýän strcat(s,t) standart functionyna seredeliň. s stringde t stringi hem özünde saklap biljek derejede orun bar diýeliň. Biz strcat functionyny hiç-hili netijäni gaýtaryp bermez ýaly edip ýazdyk. Bibliotekadaky strcat functiony bolsa netijeleýji stringe ýüzlenmäni (pointer, ссылка) gaýtaryp berýär.

```
/* strcat: t-ni s-iň soňunda ýerleşdirýär; s geregiçe uly */
void strcat(char s[], char t[])
{
    int i,j;
    i = j = 0;
    while (s[i] != '\0') /* s-iň soňuny tapýarys */
        i++;
    while ((s[i++] = t[j++]) != '\0') /* t-ni kopirleýäris */
        ;
}
```

t-den s-e indiki litera kopirlenende i we j üçin ++postfiks operator ulanylmagynyň sebäbi, gaýtalanmaklygyň her bir ädiminde i we j ululyklaryň geçirilýär literanyň ornuny dogry kesgitlemekleri üçindir.

Geronyň formulasy boýunça üçburçlygyň meýdanyny hasaplaýan programma:
#include <stdio.h>

```

#include <conio.h>
#include <math.h>
int main(void)
{
    clrscr();
    float x1,y1,x2,y2,x3,y3,a,b,c,p,s;
    printf("x1="); scanf("%f",&x1);
    printf("y1="); scanf("%f",&y1);
    printf("x2="); scanf("%f",&x2);
    printf("y2="); scanf("%f",&y2);
    printf("x3="); scanf("%f",&x3);
    printf("y3="); scanf("%f",&y3);

    a=sqrt(pow(x2-x1,2)+pow(y2-y1,2));
    b=sqrt(pow(x3-x1,2)+pow(y3-y1,2));
    c=sqrt(pow(x3-x2,2)+pow(y3-y2,2));

    p=(a+b+c);
    printf("Perimetr = %5.2f\n",p);
    p=p/2;

    s=sqrt(p*(p-a)*(p-b)*(p-c));

    printf("Meydany = %5.2f",s);

    getch();
    return 0;
}

```

Ýumuş 1. s2 strinde gabat gelýän literalaryň hemmesini s1 stringden aýyryň, squeeze(s1,s2) functionyň bir wersiýäsyny ýazmalyň.

Ýumuş 2. Eger s1 setir ululygyny düzýän literalaryň içinden, s2 setir ululygyndaky literalaryň biri bilen gabat gelýäni bar bolsa, onda onuň duran ornuny çapa çykarýän ýa-da -1-i (minus 1-i) gaýtaryp berýän (s1-iň hiç bir literasy s2-iň literalary bilen gabat gelmeýän bolsa) any(s1,s2) functiony ýazyň. (Standart bibliotekadaky strpbrk functiony edil şol işi ýerine ýetirýär, ýone literanyň nomerini dälde, görkezijisini yzyna gaýtaryp berýär.)

Bitleýin operatorlar

C dilinde bitler bilen işlemäge mümkinçilik berýän alty sany operator bar. Olary diňe bitin tipli operandlar bilen ýagny alamatly ýa-da alamatsyz char, short, int we long tipler üçin ulanmak bolar.

```

&      bitleýin AND
|      bitleýin OR

```

\wedge bitleýin öçüriji OR
 \ll çepe süýşürmek
 \gg saga süýşürmek
 \sim bitleýin inkär etmek (unar)
 $\&$ operator (bitleýin AND) razrýadlaryň toparynyň birnäçesini nula
 öwürmek üçin ulanylýar. Mysal üçin,
 $n = n \& 0177;$
 n ululykdaky kiçi ýedi bitinden galanlaryny arassalaýar.
 $|$ operator (bitler üçin OR) bitleri dikeltmek üçin ulanylýar; ýagny
 $x = x | SET_ON;$
 x ululygyň bitlerinde degişlilikde SET_ON ululykdaky bar bolan we gabat gelýän
 1-lik bitler ýazylyar.

Eýe bolmak operasiýasy we aňlatma

Eýe bolmak operatorynynda aňlatmanyň çep tarapyndaky üýtgeýän ululyk aňlatmanyň sag tarapynda hem gelýän bolsa, onda ony tygşytly görnüşde hem ýazmak bolar. Mysal üçin aşakdaky aňlatmany:

$i = i + 2;$

aşakdaky görnüşde hem ýazmak bolar.

Umumy görnüşde:

aňlatma1 **op**= aňlatma2

bu ýazgy

$aňlatma1 = (aňlatma1) \text{ op } (aňlatma2)$

Bu ýerde **op** aşakdakylaryň birine deň bolup biler

$+ \quad - \quad * \quad / \quad \% \quad \ll \quad \gg \quad \& \quad \wedge \quad |$

Mysal üçin

$x *= y + 1$

operator aşakdaky operatora ekwiwalentdir

$x = x * (y + 1)$

Bu ýerde özüniň bitin tipli argumentindäki ululygyň ikilik ýazgysyndaky birlik bitleriň sanyny kesgitlemeklige mümkinçilik berýän bitcount(x) funksiýasyna seredeliň.

/* bitcount: x-iň ikilik ýazgysyndaky birlik bitleriň sanyny kesgitleýär */

int bitcount(unsigned x)

{

int b;

for (b = 0; x != 0; x >>= 1)

if (x & 01)

b++;

return b;

}

Funksiýalary döretmek we ulanmak. Funksiýany döretmek

Kesgitleme: Funksiýa – bu programmanyň özbaşdak birligi bolup anyk meseläni görkezmek üçin ulanylýar. Si dilinde funksiýa beýleki dillerdäki ýaly podprogrammalary we proseduralary meňzeşdir.

Funksiýany çagyrmak bilen käbir hereketiň ýa-da aňlatmanyň ýerine ýetirilmegine getirýär. Funksiýany döretmäge degişli bir mysala seredeliň:

```
# include <stdio.h>
# define A “Ashgabat”
# define B “B.Annanow köç.”
# define C “TPI”
main ( )
{
TPI ( );
Printf (“%s \n”,A);
Printf (“%s \n”,B);
Printf (“%s \n”,C);
TPI ( );
}
TPI ( );
{
int a;
for (a=1; a<=65; a++);
putchar (*);
putchar ('\n');
}
Netijede
:
.....
B.Annanow köç.
TPI
.....
```

Bu programma TPI funksiýany döretdik we ulandyk. Bu funksiýa 65 sany * simwoly ekrana çykarýar.

Funksiýanyň argumenti

Ulanylýan programmanyň netijesinde ýazgyny ortadan ýerleşdireliň. Bu ýagdaýa ulanjak funksiýanyň argumentini görkezmeli bolýarys. Mysal:

```
# include <stdio.h>
```

```

# define A "Ashgabat"
# define B "B.Annanow köç."
# define C "TPI"
main ( )
{
TPI ( );
Int b;
KT (29);
Printf ("%s \n",A);
B=(65-len (B)/2);
KT(b);
Printf ("%s \n",B);
KT(b);
Printf ("%s \n",B);
KT((65-len ©)/2);
Printf ("%s \n",C);
TPI ( );
}
TPI ( );
{ int a;
for (a=1; a<=65; a++);
putchar ( '*');
putchar ( '\n');
}
KT©;
Int c;
{
int a;
for (a=1; a<=c; a++);
putchar ( ' ');
}

```

Bu programmada KT()-iň argumentini 3 usulda berildi.

1. KT (29);
2. b=(65-len (b))/2;
KT(b)ž
3. KT ((65-len (c))/2);
KT c
Int c
KT (int c)

Bu programmada A üýtgeýäni 2 ýerde ulandyk. TPI () ulanylýan A bilen KT ulanylýan A biri-biri bilen baglanşyksyz.

Return operatory

Bu operator funksiýanyň bahasyny gaýtarmak üçin ulanylýar. Şonuň absolut bahasyny gaýtarmak üçin ulanylýar. Şonuň absolut ululygyny tapmagyň funksiýasyny döredeliň. Goý, bu funksiýa ABŇ bolsun. Funksiýanyň ýerine

ýetirilişiniň netijesinde položitel ululyk alynmaly. Bu ululuk hem return operatoryň kömegi bilen gaýtarylýar. Mysal:

```
# include <stdio.h>
main ( )
{
int a=10, b=0, c=-22;
int d, e, f;
d=abs (“%d %d \n”, d,e,f);
}
abs(x);
int x;
{
int y;
y = (x<0)?-x;x;
return (y);
}
```

Netijede:

Bu programmada y üýtgeýän ()-ňiçki obýektli bolup, onuň bahasyny return operasiýanyň kömegi bilen berilýär.

```
Abs (int x)
{
if (x<0)
return (x);
else
return (x);
}
```

Adresi tapmak, & - operator

& - operator ýerine ýetirilmeginiň netijesinde degişli üýtgeýäniň huşdaky üýtgeýäniň adresi kesgitlenýär.

```
A=24
& a=12126
a=24
printf (“%d %d”, a, &a);
Netijede: 24 12126
# include <stdio.h>
main ( )
{
int a=2, b=5;
printf (“main ( )-de a=%d we &a=%d üçin”, a&a);
printf (“main ( )-de b=%d we &b=%d üçin ”, b&b);
KT (b);
}
KT (b);
{
```

```
printf ("KT ( )-de a=%d we &a=%d üçin", a&a);
printf ("KT ( )-de b=%d we &b=%d üçin", b&b);
}
main ( )-de a=2 we &a=65524
main ( )-de b=5 we &b=65522
KT( )-de a=2 we &a=65514
KT( )-de b=5 we &b=65520
```

Funksiýanyň tipini görkezmek.

Funksiýanyň tipi argumenti bilen bagly dälde, öz bahasy bilen kesgitlenýär. Eger funksiýanyň tipi görkezilmese, onda ol funksiýa int görnüşe degişlidir. Ýönekeý funksiýalara seredeliň.

```
Float KT (x,y);
```

```
Int x,y;
```

```
{
int z;
z=x/y;
return (z);
}
```

Mysal:

S,t hakyky san berlen. Tapmaly:

$f(t, -2s, 1.17) + f(22, t, s-t)$ nirede $f(a, b, c) = 2a - b - \sin c / 5 + |c|$

```
# include <stdio.h>
```

```
# include <math.h>
```

```
main ( )
```

```
{
float a, b, c, st, d, e.f();
printf ("s="); scanf ("%f",&s);
printf ("%f",&s);
a=t; b=-2*s; c=1.17, d=KT(a,b,c);
a=2.2; b=t; c=s-t;
d=KT(a,b,c);
printf ("f ( )=%f\n", d+e);
}
```

```
float KT (a,b,c);
```

```
float a,b,c;
```

```
{
float z;
z=(2*a-b-sin c)/(5+tabs (c));
return (z);
}
```

Onlyk sanawyş ulgamyndaky(s/u) sany ikilik s/u-na geçirýän programma:

```
#include <stdio.h>
```

```
#include <conio.h>
```



```

int bitcount(unsigned x);
//-----//
main()
{ clrscr();
  unsigned char c;
  while((c = getche()) != 27)
  {
    printf(" --> ");
    bitcount(c);
  }
}
//-----//
/* x sany ikilik sanaýyş sistemasyna geçirýar */
int bitcount(unsigned x)
{
  int b,s[8];
  for(b=1; b <= 8; b++,x >>= 1)
    s[b]=(x & 01);
  for(b=8; b>=1; b--)
    printf("%d",s[b]);

  printf("\n");
  return b;
}

```

Funksiýalar, funksiýanyň täsir edýän çägi we ýadyň klaslary

Huşň klasy üýtgeýäniň hereketlenen ýaýlasyny we bolmaklygyň dowamlylygyny kesgitleýär. Üýtgeýäniň degişli tipiniň boluşy ýaly degişli huşň klasy hem bolmalydyr.

Huşň klasy aşakdaky görnüşe eýedir.

№	Huşň klasy	Ýazylyşy	Dowamlylygy
1.	Awtomatik	awto	wagtlaýyn
2.	Registrlı	register	wagtlaýyn
3.	Statistiki	static	hemişelik
4.	Daşky	extern	hemişelik
5.	daşky statistiki	statik	hemişelik

1. Awto – eger üýtgeýäniň önünde huşň klasy görkezilmese, onda ol huşň awtomatiki klasyna degişlidir. Bu üýtgeýän funksiýanyň içinde gelmelidir. Mysal:

```

#include <stdio.h>
main ( )
{
  awto int a;
  ...

```

```
}
```

2. Registor – registrleri üýtgeýän merkezi prossesoryň registrde ýerleşip, ol huşdaka görä çagyrmaklygy we işi ýerine ýetirmekligi çalt amala aşyrylýar. Köp halatlarda bolsa, huşuň awtomatiki klasy meňzeşdiler. Onuň bolmaklygynyň dowamlylygy wagtlaýyndyr. Mysal:

```
# include <stdio.h>
main ( )
{
register int a;
...
}
```

3. Static – statistiki üýtgeýän awtomatiki üýtgeýäne meňzeşdir, ýöne onuň bolmaklygynyň dowamlylygy hemişelikdir. Huşuň bu klasyna berlen üýtgeýän funksiýanyň içinde berilmelidir. Mysal:

```
# include <stdio.h>
main ( )
{
int a;
for (a=1; a<=3; a++);
{
printf (“Mysal %d”, a);
KT ( );
}
}
KT ( );
{
int b=1;
static int c=1;
printf (“b=%d we c=%d \n”, b++, c++);
}
```

Netijede:

Mysal 1: b=1 we c=1

Mysal 2: b=2 we c=2

Mysal 3: b=3 we c=3

4. Daşky üýtgeýän extern-huşuň daşky üýtgeýän klasyna degişli bolan üýtgeýän funksiýanyň öňünden berilýär. Bu daşky üýtgeýäniň bolmaklygyň dowamlylygy hemişelikdir.

```
# include <stdio.h>
int a;
char ch;
main ( )
{
extern int a;
extern char ch;
}
```

Eger programmada ýeke funksiýa ulanylýan bolsa, onda extern sözünü ýazmaklygyň zerurlygy ýok. Eger birnäçe funksiýa bolsa, onda hökmany görkezilmeli.

5. Daşky statistiki – huşuň statistiki klasyna degişli bolan üýtgeýän hem funksiýanyň önünden berilýär. Daşky üýtgeýän bilen daşky statistiki üýtgeýäniň tapawudy. Daşky üýtgeýäni birnäçe faýlda hem ulanyp bolýar, ýöne daşky statik üýtgeýän diňe bir faýla degişlidir. Mysal:

```
#include <stdio.h>
static int a;
static char ch;
static main ( )
{
    static int a;
    static char ch;
}
```

Si-de esasy ulanylýan huşuň klasy awtomatiki görünüşdir.

Faktorialy hasaplaýan programma:

```
#include <stdio.h>
#include <conio.h>
long fac(long n);
main(void)
{
    clrscr();
    long a=10;
    printf("%ld\n",fac(a));
    getch();
    return 0;
}
//-----//
long fac(long b)
{
    long p=1; long i;
    for(i=1; i<=b; ++i) p=p*i;
    return p;
}
```

```
#include <stdio.h>
#include <conio.h>
void md(int a[], int n);
void main()
{
    clrscr();
    int i, x[5];
```

```

md(x,4);
printf("-----\n");
for(i=0; i<=4; ++i)
    printf("x[%d]=%d\n",i,x[i]);
}
/*-----*/
void md(int a[], int n)
{
    int k;
    for(k=0; k<=n; ++k)
        { printf("a[%d]=",k); scanf("%d",&a[k]); }
}

```

Maglumatlaryň şahalaýyn gurluşy

Şol bir tipe degişli bolan elementleriň toplumyna massiwler diýilýär. Olar aşakdaky ýaly beýan edilýär. Mysal:

```

int a[20]
a[0], a[1], ..., [19]
a[0]=1 a[5]=5
float b[10]
b[3]=5.25
b[6]=2.3

```

Mysal: Islendik 10 sanyň orta bahasyny tapmaklygyň programmasyny düzmeli.

```

#include <stdio.h>
#define num 10
main ( )
{
    int i, a, s, c[num];
    for (i=0; i<num; i++);
    scanf ("%d", & c[i]);
    for (i=0; i<num; i++);
    printf ("%d", c[i]);
    printf ('\n');
    for (s=0; i=0; i<num; i++);
    s+=c[i];
    a=s/num;
    printf ("orta baha = %d", a);
}

```

Massiwler huşun degişli klasyna degişli bolup durýar. Mysal:

```

#include <stdio.h>
int a[20];
main ( )
{
    float b[30];
    static char ch[10];
}

```

```
extern int a[ ];
```

Mysal: Her aýyň degişli günleri ekrana çykarmagyň programmasyny düzeliň.

```
# include <stdio.h>
int gun[12]={ 31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 }
main ( )
{
int a;
extern int gun [ ];
for (a=0; a<12; a++);
printf ("%d-ay, %d-gun \n", a=1, gun[a]);
```

Görkeziji operasiýasy.

Görkeziji operasiýasy üýtgeýäniň simwoliki bahasyny görkezýär. * görkeziji operasiýasy. Mysal:

```
A=2;
B=&a;
C=*b;
C=2;
# include <stdio.h>
main ( )
{
int a[4]*b, c;
float a[4]*a;
b=a; (b=2a[0]);
e=d; (e=&d[0]);
for (c=0; c<4; c+1)
printf ("görkeziji %d u we %d u", c, b+c, e+c);
}
```

Netijede: görkeziji: 0; 65518 we 65498

1; 65580 we 65502

2; 65522 we 65506

3; 65524 we 65510

Mysal1: Matrisalaryň jemini tapar ýaly programma düzmeli.

1 2 5 1 6 3

+ =

3 4 0 2 3 6

```
# include <stdio.h>
```

```
# include <math.h>
```

```
main ( )
```

```
{
```

```
int i, j, c[2][2];
```

```
static int a[2][2]={ 1,2,3,4};
```

```
static int b[2][2]={ 5,1,0,2};
```

```
for (i=0; i<2; i++); { for (j=0; j<2; j++)
```

```

c[i][j]=a[i][j]+b[i][j];}
for (i=0; i<2; i++) {for (j=0; j<2; j++)
printf ("\n c[%d, %d]=%d", i+1, j+1, c[i][j]);
}}

```

Mysal2: Matrisalaryň köpeltmek hasyly alynar ýaly programma düzmeli.

```

1 2   5 1   5 5

```

```

*      =

```

```

3 4   0 2   15 11

```

```

#include <stdio.h>

```

```

#include <math.h>

```

```

main ( )

```

```

{

```

```

int i, j, c[2][2];

```

```

static int a[2][2]={ 1,2,3,4};

```

```

static int b[2][2]={ 5,1,0,2};

```

```

for (i=0; i<2; i++); {for (j=0; j<2; j++){

```

```

c[i][j]=0;

```

```

for (k=0; k<2; k++)

```

```

c[i][j]=a[i][k]*b[k][j];} }

```

```

for (i=0; i<1; i++) {for (j=0; j<2; j++)

```

```

printf ("\n g[%d, %d]=%d", i+1, j+1, c[i][j]);

```

```

}}

```

```

#include <stdio.h>

```

```

#include <conio.h>

```

```

#include <stdlib.h>

```

```

void mg(int d);

```

```

void main()

```

```

{

```

```

clrscr(); randomize();

```

```

int i;

```

```

extern float x[10];

```

```

mg(10);

```

```

for(i=0; i<10; ++i)

```

```

printf("a[%d]=%3.1f\n",i,x[i]);

```

```

getch();

```

```

}

```

```

/*-----*/

```

```

float x[10];

```

```

void mg(int m)

```

```

{

```

```

int i;

```

```

for(i=0; i<m; ++i)

```

```

x[i]=random(100)*0.2;

```

}

Simwol setirli char tipe degişli massiw bolup soňky ‘10’ simwol bilen gutarýar. Simwol setiri girizilende köp halatda huşuň ölçegi anyk görkezilýär. Mysal üçin: `char a[20] = “bahar geldi”`. Bu ýagdaýda huşda 20 öýjük goýulýar we onuň 12-si simwollar bilen doldurylýar. Galanlary bolsa, ýagny galan 8 öýjük ‘10’ simwoly bilen doldurylýar. Simwol setiri hökmany ýagdaýda huşuň statistiki ýa-da daşky klasyna degişli bolmalydyr.

Eger static `char a[] = { ‘B’, ‘a’, ‘h’, ‘a’, ‘r’, ‘ ‘, ‘g’, ‘e’, ‘l’, ‘d’, ‘i’, ‘.’, ‘%’ }`

Eger setir döretjek bolsa, onda * operasiýasyny ulanmaly. Mysal üçin: static `char *a = “Bahar geldi”`. Bu ýagdaýda a massiwiň bir elementi bar, ol hem bir setirden ybarat. Eger a massiwiň simwoly birden köp bolsa, onda ol görkezilýär. Mysal üçin: static `char *a[3] = { {salam}, {sag bol}, {bahar geldi} }`; Bu ýerde a [0] elementde “salam”, a[1] elemente “sag bol”, a[2] elementde “bahar geldi”, degişlidir. Setirleriň ölçegini anyk görkezip hem bolýar. Mysal üçin: 1. static `char a[3][8]`

2. static `char *a[3]`

Bularyň tapawudy birinji ýagdaýda her setirdäki simwollara 8 öýjik goýulýar, ýöne ikinji ýagdaýda gerekli ýer goýulýar. Ýagny ikinji ýagdaýda artykmaç ýer eýelenmeýär. Setirleri girizmek we çykarmak üçin degişlilikde `gets()` we `puts()` funksiýalary ulanylýar.

1. `gets ()` funksiýasy – bu funksiýa setiri girizmek üçin ulanylýar. Ol iki ädime amala aşyrylýar.

1. huşda ýer bellemek

2. funksiýa ulanmak

Huşda ýer bellemek bolanda giriziljek setiriň ölçegini bermekdir. Mysal üçin `char a[81]` funksiýa ulanmakda esasan `gets()` funksiýasy ulanylýar. Bir mysala seredeliň:

```
# include <stdio.h>
main ( )
{
static char a[81];
printf (“Adyň näme?\n”);

gets (a);
printf (“%s gowy ad\n”, a);
}
```

Netijede:

Adyň näme?

Myrat

Myrat gowy ad

`Gets ()` funksiýa bilen `scanf` funksiýanyň tapawudy `scanf ()` funksiýasy % format bilen söz okaýar. Ýöne `gets ()` bolsa, tutuş setiri okaýar.

2. `Puts` funksiýasy – kabul edilen setiri çapa çykarýar. Argumentinde çapa çykmaly setiri görkezmeli. Mysal:

```
# include <stdio.h>
```

```

main ( )
{
static char a[ ]= "bahar geldi";
static char *b= "gyş geldi";
puts ("ýagyş ýagýar");
puts (a);
puts (b);}
Netijede:   ýagyş ýagýar
            bahar geldi
            gyş geldi

```

printf () bilen puts () funksiýanyň tapawudy: puts () funksiýasy her setiri awtomatik täze setirden ýazýar. Ýöne printf ()-de bolsa '\n' simwoly ýazmaly bolýarys. Setirler bilen esasan dört sany funksiýa işleýär: strlen (), strcat (), strcmp (), strcpy ().

1. strlen () – bu funksiýa setiriň uzynlygyny kesgitleýär. Mysal:

```

#include <stdio.h>
main ( )
{
statis char *a = "bahar geldi";
printf ("%d \n", strlen (a));
}

```

2. strcat () – bu funksiýa iki setiri birleşdirmek üçin ulanylýar. Mysal:

```

#include <stdio.h>
main ( )
{
statis char *b = "ýagdaýlaň gowymy?";
puts ("adyň näme?");
gets (a);
strcat (a,b);
puts (a);
}

```

Netijede: Adyň näme? Bahar. Bahar ýagdaýlaň gowymy?

3. strcmp () – bu funksiýa setirleri deňeşdirmek üçin ulanylýar. Mysal:

```

#include <stdio.h>
#define A "Bahar"
main ( )
{
static char a[40]
puts ("Adyň näme?");
gets (a);
while (strcmp (a, A)!=0);
{
puts ("ýalňyş");
gets (a);
}
}

```



```
puts ("dogry");
}
```

4. strcpy () – bu funksiýasy setiriň kopiýasyny almak üçin ulanylýar. Mysal:

```
# include <stdio.h>
# define A "Ýalňys"
main ( )
```

```
{
static char *a=A;
static char b[40]
puts (a);
puts (b);
strcpy (b, a);
puts (a);
puts (b);
Netijede: ýalňys barla
         ýalňys barla
         ýalňys barla
```

C++-de giriş çykyş

Si diliniň kitaphanasynda üç ýol bilen girip bolýar.

1. Awtomatiki
 2. Faýla geçmek ýoly (# include)
 3. Kitaphana giriş (lc.lib)
- Kitaphanadan biziň ulanan funksiýalarymyz:

1. Giriş-çykyş funksiýalary

```
getchar( ) – simwol almak
putchar( ) – simwol çap etmek
gets ( ) – setiri almak
puts ( ) – setiri çap etmek
scanf – formatirlenen giriş
printf – formatirlenen çykyş
```

2. Setir bilen işleýän funksiýalar.

```
Strlen ( ) – setiriň uzynlygy
Strcat ( ) – setirleri birleşdirmek
Strcmp ( ) – setirleri deňeşdirmek
Strcpy ( ) – setirleriň kopiýasyny almak
```

3. Faýyllary açmak – ýapmak unksiýasy.

```
Fopen ( ) – faýly açmak
Fclose ( ) – faýly ýapmak
Gets ( ) – faýyldan simwol almak
Puts ( ) – faýlda simwoly çap etmek
```

1) Fopen () funksiýasy – bu funksiýa faýly açmak üçin ulanylýar. Fopen () funksiýasy üç sany esasy parametri dolandyryar:

- a) Faýlyň ady
- b) Faýly nähili ulanjak:

- “r” – okamak
- “w” – ýazmak
- “a” – goşmak

Bellik: eger bar bolan faýla “w” ulansak, onda öňki ýazgylar ýitip, täze ýazgylar galýar.

c) Faýly görkeziji bolup durýar.

FILE in:

In = fopen (“ady”, “r”);

Bu ýerde in “ady” diýlen faýla görkeziji bolup durýar.

2) Fclose () – bu funksiýa faýly ýapmak üçin ulanylýar. Fclose (in); ulanylýar. Bu ýerde argumentde faýlyň görkezijisi ýazylmaly.

3) Gets () – Puts () faýyllary girizmek we çykarmak.

Ch=gets (in); - bu görkezijidäki faýldan simwol almak.

Puts (ch, out); – bu ch simwoly faýla ýazmak.

Puts (ch, stdout); - simwoly standart çykyşa ibermek. Mysal:

```
#include <stdio.h>
main ( )
{
FILE *in;
Int ch;
If ((in=fopen (“ady”, “r”))!=NULL)
{
while ((ch=gets (in)!=EOF)
puts (ch, stdout);
fclose (in);
}
else
printf (“faýl açylmaly \n”);}
```

mesele:

n sany otag bar. Her otagyň gapysynda onuň tertip boýunça ýazylan(N1, N2, ..., Nn). Otaglarda çyra bar. Ol çyra şol otagdaky açaryň kömegi bilen ýakylýar ýa-da öçürilýär. Her otagdaky açaryň iki sany durnukly ýagdaýy bar: 0,1; 0 ýagdaýda otagdaky çyra öçýär, 1 ýagdaýda bolsa otagdaky çyra ýanýar. Açaryň ýagdaýy onuň düwmejigine basylmak bilen üýtgedilýär. Başlangyç ýagdaýda hemme otaglardaky çyralar öçürilen. Ilkinji okuwçy hemme otaglara girip açarlary basyp çykýar. Soňra bolsa ikikji okuwçy her ikinji otaga girip, üçünji okuwçy bolsa her bir üçünji otaga girip we ş.m., n-nji okuwçu bolsa diňe n-nji otagdaky açaryň ýagdaýyny üýtgedip gaýdýar. Iň soňunda näçe otagda çyra ýanyk galar? Bu meseläni çözmek üçin programma ýazmaly.

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
```

```

main()
{
    clrscr();
    int a[1001],i,j,n,s=0;
    cout<<"n="; cin>>n;
    for(i=1; i<=n; i++) a[i]=0;

    for(i=1; i<=n; i++)
        for(j=i; j<=n; j=j+i)
            if(a[j]==0) a[j]=1; else a[j]=0;

    for(i=1; i<=n; i++)
        if(a[i]>0) s++;

    printf("s=%d",s);
    getch();
}

```

mesele:

Temperaturanyň feregeýtdäki bahasyny selsä geçirmek üçin programma ýazmaly

```
#include <stdio.h>
```

```
#include <conio.h>
```

```
main()
```

```
{
```

```
    clrscr();
```

```
    float fahr, celsius;
```

```
    int lower, upper, step;
```

```
    lower=0;
```

```
    upper=300;
```

```
    step=20;
```

```
    celsius=lower;
```

```
    printf("Celsius \t Fahr\n");
```

```
    printf("-----\n");
```

```
    while( celsius <= upper )
```

```
    {
```

```
        fahr=(9.0)/(5.0)*celsius+32;
```

```
        printf("%6.0f\t%3.1f\n",celsius,fahr);
```

```
        celsius = celsius + step;
```

```
    }
```

```
    getch();
```

```
}
```

C++ dili öz ülanyjylaryna diňe bir programma ýazmaklyga däl-de eýsem 300-den gowrak biblioteka functionlary hem-de macrolary ulanmaklyga

mümkinçilik berýär. Şeýle macrolaryň we functionlaryň köpüsi C diliniň dürli görnüşleri bilen ylalaşykly ulanylyp biliner.

C++ diliniň hemme bibliotekalary üçin interface faýly döredilendir(.h giňeldilmeli faýllar) döredilendir. Ol faýllarda hemme biblioteka functionlarynyň prototipleri saklanýarlar.

Mesele:

girizilen tekstde näçe sany setiriň, boşlugyň, sifriň bardygyny hasaplaýan programma ýazmaly:

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    clrscr();
    int c, i, nwhite, nother;
    int ndigit[10];
    nwhite = nother = 0;
    for (i=0; i<10; ++i) ndigit[i]=0;
    while((c=getch()) != 27)
        if (c>='0' && c<='9') ++ndigit[c-'0'];
        else if (c==' ' || c=='\t') ++nwhite;
        else ++nother;
    printf("numbers=");
    for (i=0; i<10; ++i) printf(" %d",ndigit[i]);
    printf("\nspace=%d other=%d\n",nwhite, nother);
    getch();
    return 0;
}
```

mesele:

Bitin sany bitin derejä götermäge mümkinçilik berýän programma ýazmaly

```
#include <stdio.h>
#include <conio.h>
int power(int m, int n);
main(void)
{
    clrscr();
    int i;
    for(i=0; i<10; ++i)
        printf("%d\t%d\t%d\n",i,power(2,i),power(-3,i));
    getch();
    return 0;
}
int power(int base, int n)
{
```

```

int i,p;
p=1;
for(i=1; i<=n; ++i)
    p=p*base;
return p;
}

```

mesele:

Girizilen setiriň içinden iň uzynyny kesgitlemäge mümkinçilik berýän programma:

```

#include <stdio.h>
#include <conio.h>
#define MAXLINE 1000
int getline(char line[], int maxline);
void copy(char to[], char from[]);
main(void)
{
    int len;
    int max;
    char line[MAXLINE];
    char longest[MAXLINE];
    clrscr();
    max=0;
    while ((len=getline(line, MAXLINE)) > 0)
        if (len > max)
        {
            // printf("%s --> %d\n",line,len);
            max=len;
            copy(longest, line);
        }
    if(max>0) printf("%s\n",longest);

    getch();
    return 0;
}
//-----//
int getline(char s[], int lim)
{
    int c,i;
    for(i=0; i<lim-1 && (c=getche()) != 13; ++i)
        s[i]=c;

    if (c=='\n')
    {
        s[i]=c; ++i;
    }
}

```

```

    }
    s[i]='\0';
    printf("\n");
    return i;
}
//-----//
void copy(char to[], char from[])
{
    int i;
    i=0;
    while((to[i] = from[i]) != '\0')
        ++i;
}

```

C++ diliniň ýörite serişdeleri

C diliniň preprocessor kompilýatora berilmezden öň, tipi *.cpp bolan, tekst faýllary işlýär. Preprocessor hemme macroçagyryşlary giňeltýär we daşky faýllary ýerine goýýär.

Preprocessorň komandalary

Preprocessorň komandasyny aňlatmak üçin # belgi ulanylýar. Bu simwili C dilinde başga maksat üçin ulanylmaýar. Preprocessorň komandalarynyň sanawy:

komanda	beýany
#define	makrony kesgitlemek
#undef	Makronyň kesgitlemesini inkär etmek
#include	daşky faýldan tekst goýmak
#if	aňlatma hemişeligine baglylykda teksti şertli goýmak
#ifdef	Makronyň kesgitlenmegine baglylykda tekstiň bölegini goýmak
#ifndef	makro kesgitlenmedik bolsa tekst bölegini goýmak
#else	#ifdef we #ifndef üçin alternatiwa
#endif	şertli goýulýan tekst böleginiň soňyny kesgitlemek
#ilne	setiriň nomerini görkezýär
#defined	#if bilen bilelikde ulanylýar. Görkezilen adyň makronyň adydygyny ýa-da dældigini kesgitleýär.
#pragma	kompilýator ýerine ýetýän döwründe görkezme berýär
#error	kompilýasiýa döwründe ýüze çykýan ýalňyşlyklar barada habary ýüze çykarýar

C++ dilinde C diliniň täze ANSI standarty kesgitlenenidir. Diliň bu görnüşünde köp täzelikler girizilendir. Umuman programmirleme dilleri aşak derejeli hem-de ýokary derejeli diýilip atlandyrylýan iki topara bölünýärler. C dili bolsa olaryň ikisine hem degişli bolan ortalık bir dil bolup onuň mümkinçilikleri juda uludyr.

maglumatlaryň tipleri:

Tipler	Uzynlygy	Çägi
unsigned char	8 bits	0 to 255
char	8 bits	-128 to 127
enum	16 bits	-32,768 to 32,767
unsigned int	16 bits	0 to 65,535
short int	16 bits	-32,768 to 32,767
int	16 bits	-32,768 to 32,767
unsigned long	32 bits	0 to 4,294,967,295
long	32 bits	-2,147,483,648 to 2,147,483,647
float	32 bits	$3.4 * (10^{*-38})$ to $3.4 * (10^{*+38})$
double	64 bits	$1.7 * (10^{*-308})$ to $1.7 * (10^{*+308})$
long double	80 bits	$3.4 * (10^{*-4932})$ to $1.1 * (10^{*+4932})$

sys_errlist massiwiniň elementleri we olaryň aňladýan manylary:

Elemanti	Manysy
E2BIG	Arg list too long
EACCES	Permission denied
EBADF	Bad file number
ECONTR	Memory blocks destroyed
ECURDIR	Attempt to remove CurDir
EDOM	Domain error
EEXIST	File already exists
EFAULT	Unknown error
EINVACC	Invalid access code
EINVAL	Invalid argument
EINVDAT	Invalid data
EINVDRV	Invalid drive specified
EINVENV	Invalid environment
EINVFMF	Invalid format
EINVFNCF	Invalid function number
EINVMEM	Invalid memory block address
EMFILE	Too many open files
ENMFILE	No more files
ENODEV	No such device
ENOENT	No such file or directory
ENOEXEC	Exec format error
ENOFILF	No such file or directory
ENOMEM	Not enough memory

ENOPATH	Path not found
ENOTSAM	Not same device
ERANGE	Result out of range
EXDEV	Cross-device link
EZERO	Error 0

2. C++ PROGRAMMIRLEME DİLİNİN ESASLARY. VISUAL PROGRAMMIRLEME GURŞAWLARY

2.1. .NET Framework gurşawy

Windows operasion ulgamyna laýyk (adekwat) programmirmek asla kyn dälär. Hakykatdan Microsoft Visual C++ 2010 işläp düzülýän gurşawy bu prosesi örän ýenilleşdir ýar, şu kitaby okanyňyzda siz muňa göz ýerirersiniz. Bu ýolda diňe bir pasgelçilik bar: Windows programmirmeginiň aýratynlygyna ýüzlenmezden ön siz C++ 2010 programmirme diliniň serişdeleri bilen, hususan-da, onuň obýekte gözükdirilen aspektleri (jähtleri) bilen gowy tanyş bolmalysynyz. Obýekte gözükdirip programmirmeginiň tehnika Windows üçin programma ýazylanda peýdalanylýan Visual C++ 2010 gurşawynyň ähli gurallarynyň netijeliliginiň açary bolup dur ýar, şonuň üçin hem siziň ona gowy düşünmeginiz möhümdir. Şu kitabyň maksady hem sondan ybaratdyr.

Şu bap C++ dilinde goşundylary programmirmäge degişli möhüm konsepsiýalara syny tekli ed ýar. Biz sizi işläp düzmegiň integrirlenen gurşawy bilen (Integrated Development Environment - IDE) gysgaça tanyşdyrarsy, ol Visual C++ 2010 işläp düzüş gurşawy bilen bilelikde üpjün edil ýar. Bu gurşaw özüniň ähli amallarynda yönekeýdir we intuitiw düşnükli, sonun üçin hem ondan peýdalanyň başlamak bilen siz onun mümkinçilikleriniň köpüsini tiz ele alyp bilersiniz, işläp düzüş gurşawyny öwrenmäge in gowy çemeleşme yönekeý programmany döretmek, kompilirmek we ýerine ýetirmek prosesleriniň ählisini geçirmekden ybaratdyr.

Indi bolsa öz kompýuterleriňizi işe girizip, Windowsy açyňda, işläp düzmegiň Visual C++ 2010 kuwwatly gurşawyny işe girizmek bilen gyzykly syýahata başlaň.

.NET Framework gurşawy - Microsoft korporasiýasynyň beýleki .NET işläp düzüş serişdeleri ýaly - işläp düzmegiň Visual C++ 2010 gurşawynyň merkezi bölegi bolup durýar. .NET Framework gurşawy iki elementden, ýagny siziň programmaňyzyň ýerine ýetirilýän umumy dilli ýerine ýetiriji gurşawyndan (Common Language Runtime - CLR) we .NET Framework kitaphanalaryň klaslary diýlip atlandyrylýan kitaphanalar toplumyndan durýar. .NET Framework kitaphanalar klasy funksional goldawy amala aşyr ýar, bu goldaw ulanylýan programmirme diline garamazdan, CLR gurşawynyň dolandyryşy astynda kod üçin zerurdyr, sonun üçin hem C++, C# dilinde ýa-da .NET Framework gurşawyny goldaýan islendik beýleki dilde ýazylan .NET programmalary şol bir NET kitaphanalaryndan peýdalanylýarlar.

C++ goşundylarynyň Visual C++ 2010 gurşawyndan işläp taýýarlap bolýan biri-birinden düýpli tapawutlanýan iki görnüşi bardyr. Şular ýaly programmlar C++ diliniň ISO/IEC (International Standards Organization/International Electrotechnical Commission) standarty bilen kesgitlenen wersiýasynda ýazyl ýar. Şeýle hem siz C++ gurşawynyň dolandyrmagy astynda ýerine ýetirilýän we C++ diliniň C++/CLR adyny alan giheldilen wersiýasynyň kömegi bilen durmuşa geçirilen programmalaryny işläp düzüp bilersiňiz. Bu programmalary biz CLR programmalary ýa-da C++/CLR programmalary diýip atlandyrsy.

.NET Framework gurşawy işlap düzmegin Visual C++ 2010 gurşawynyň bölegi bolup durmaýar-da, aglaba ýagdaýda Windows amal ulgamynyň goşundylaryň ýazylmagyny we web gulluklarynyň döredilmegini üpjün edýän komponenti bolup durýar. .NET Framework gurşawy koduň we howpsuzlygyň ygtybarlylygy, şeýle hem siziň C++ koduhyzyň .NET Frameworka gözükdirilen beýleki yigrimiden gowrak programmirleme dillerinde ýazylan kodlar bilen itegrilenmek mümkinçiligi babatynda degerli artykmaçlyklary berýär. Öndüriligiň biraz peselmegi (köplenç ýagdaýda bu asla belli hem bolmaýar) .NET Frameworka gözükdirmeginiň kabir ýetmezçiligi bolup durýar.

2.2. Umumy dilli ýerine ýetiriji gurşaw (CLR)

Umumy dilli ýerine ýetiriji gurşaw (Common Language Runtime - CLR) -bu Visual Basic, C# we elbetde C++ dillerini goşmak bilen, ýokary derejeli dilleriň giň diapozonynda ýazylan programmalaryň ýerine ýetirilýän standart gurşawydyr. Häzirki wagtda CLR gurşawynyň spesifikasiýasy umumy dil infrastrukturasyň (Common Language Runtime - CLI) ECMA (European Association for Standardizing Information and Computer Systems – Ýewropanyň maglumat we hasaplaýyş ulgamlaryny standartlaşdyrmak boýunça assosiasiýasy) standartynyň, şeýle hem şonuň ýaly ISO-ISO/IEC 23271 standartynyň içinde gurlandyr, şonuň üçin hem CLR gurşawy bu standartyň durmuşa geçirilmegi bolup dur ýar. Şol sebäbe göre hem CLR gurşawy üçin C++ dili C++/CLI diýlip atlandyryl ýar - bu CLI infrastrukturasy üçin C++ dilidir, şonuň üçin siziň belli bir wagtdan soň CLI infrastrukturasyň durmuşa geçirýän beýleki amal ulgamlary üçin C++/CLI kompilyatorlary bilen iş salyşmaly bolmagyňyz mümkindir.

CLI infrastrukturasy - bu hakykat yüzünde, wirtual maşyn gurşawynyň spesifikasiýasy bolup, ol programmirlemegin köp dürli ýokary derejeli dillerde ýazylan goşundylaryň original başlangyç kody üýtgetmezden we gaýtadan kompilyasiýa etmezden dürli ulgamlarda ýerine ýetirilmegine mümkinçilik berýär. CLI infrastrukturasy wirtual maşynyň aralyk diliniň standartyny spesifisirleýär, bu standartda bolsa programmirlemegin ýokary derejeli diliniň başlangyç kody kompilirlenýar. .NET Framework gurşawynda bu aralyk dil *Microsoft* (Microsoft Intermediate Language (MSIL) *aralyk dili* diýlip atlandyrylýar. Bu aralyk diliň kody ahyr netijede programma ýerine ýetirilýän wagtynda operatiw kompilyatoryň (just-in-time - JIT) kömegi bilen maşyn koduna öwrülýär. Elbetde, CLI infrastrukturasyň aralyk dilinde kod dine CLI infrastrukturasyň amal edilyän gurşawynda hereket edip biler.

Şeýle hem CLI infrastrukturasy tiplerin umumy ulgamy (Common Type System - CTS) diýip atlandyrylýan maglumatlaryň tipleriniň umumy toplumyny kesgitleýär, bu toplum CLI infrastrukturasyň amal edilmegine gözükdirilen islendik dilde ýazylan programmalar tarapyndan peýdalanylmalydyr. CTS ulgamy maglumatlar tipleriniň CLR gurşawynda ulanylmagyny spesifisirleýär we önünden îsgitlenen tiplerin toplumyny öz içine alýar. Şeýle hem siz özünizin maglumat iplerinizi kesgitläp bilersiniz, yöne olary kesgitlemek olaryň CLR gurşawynyň -esifikasiýasy bilen laýyk bolmagy üçin birnäçe

kadalara tabyn edilmelidir. Maglumatlary görkezmek üçin tipleriň standartlaşdyrylan ulgamynyň bolmagy dürli dillerde ýazylan komponentlere maglumatlary bir ölçegli usulda işlemäge mümkinçilik beryär we dürli dillerde ýazylan komponentleriň bir goşunda igrirlenmek mümkinçil igini üpjün edýär.

Maglumatlaryn howpsuzlygy we programmalaryn ygtybarlylygy belli bir derejede bu ýadyň hereketli bölünmeginiň we boşadylmagynyň doly awtomatlaşdyrylmagy, belli bir derejede bolsa MSIL programmalarynyň kodunyň programma ýerine ýetirilmezinden ön doly werifikasiýa edilmegi netijesinde CLR gurşawy bilen ep-esli gineldildi. CLR gurşawy - bu Microsoft Windows amal ulgamynyň dolandyryan kompýuterinde hereket edýän CLI infrastrukturasynyň spesikasiýasynyň diňe bir durmuşa geçirilişidir. Beýleki amal ulgamlarynyň we apparat platformalary üçin CLI infrastrukturasynyň beýleki durmuşa geçirilişleriniň hem peýda boljakdygyna şübhe bolup bilmez. Sizin kähallatlarda CLI we CLR adalgalarynyň biri-birini çalyşýandygyna, yöne olaryn şol bir zat dældigine üns beren bolmagyňyz mümkin. CLI - bu standartyn spesifikasiýasydyr. CLR bolsa Microsoft korporasiýasynyň CLI infrastrukturasynyň durmuşa geçirilmegidir.

2.3. C++ goşundylaryny ýazmak

Visual C++ 2010 işlap düzüş gurşawy goşundylaryn dürli tiplerini we programma komponentlerini işlap düzmek babatynda adatdan daşary çeyeligi üpjün edýär. Şu bapda eýýäm CLR gurşawynda ýerine ýetirilyän kody hem, amal ulgamynyň «öz» maşyn koduna gönüden göni kompilirlenýän koduny hem ýazyp bilersiňiz. CLR gurşawyna gözükdirilen Windows goşundylary üçin GUI (Graphical User Interface - peýdalanyjynyň grafiki interfeýsi) gurmak üçin binýat hökmünde Windows Forms karkasy peýdalanylýar, ol .NET Framework binýatlyk klaslaryň kitaphanasynda berilyär. Windows Forms karkasynyň peýdalanylmagy GUI tiz işlenip düzülmegini üpjün edýar, çünki siz ony grafika görnüşinde standart komponentlerden düzyärsiňiz we doly awtomatiki usulda döredilen kody alýarsyňyz. Sonra dine zerur bolan funksionallygy gazanmak üçin bu kody biraz sazlamaly bolar.

Ýerine ýetirilyän «öz» koduny almak üçin hem birnäçe usul bardyr. Olaryň Windows goşundysyny peýdalanyjynyň grafiki interfeýsini programmirmek üçin Microsoft Foundation Classes (MFC) klaslarynyň kitaphanasyny peýdalanmakdyr. MFC kitaphanasy GUI döretmek we dolandyrmak babatynda Windows (Windows API) amal ulgamynyň programmalaýyn interfeýsini inkapsulirleýär we programmalary işlap düzmek işini ep-esli ýenilleşdirýär. API Windows interfeýsi C++ yüze çykamazdan has ön peýda boldy, sonun üçin hem onuň hiç hili obýekte gözükdirilen häsiýetnamasy ýokdur, eger ol biziň günlerimizde işlenip düzülen bolsa, onda şeýle häsiýetnamalary bolardy. Emma siziň MFC kitaphanasyny peýdalanmagyňyz hökmany daldır. Eger size öndürijilikde utuş gazanmak gerek bolsa, onda özüňiziň C++ koduňyzdan gönüden göni API Windows interfeýsiniň funksiýalaryna yüzlenip bilersiniz.

CLR gurşawynyň dolandyrmagynda ýerine ýetirilyän C++ kody C++ dolandyrylýan kod diylip atlandyrylýar, sebäbi maglumatlar we kod CLR gurşawynyfi gözegçiliginde bolýar. CLR programmalarynda maglumatlary ýerleşdirmek üçin hereketli bölünip aýrylan ýady boşatmak awtomatiki usulda amala aşyrylýar, bu bolsa C++ «öz» goşundylarynyň ýalnyşlarynyň esasy çeşmesini aradan aýyrmaga mümkinçilik beryär. CLR gurşawyndan daşarda ýerine ýetirilyän C++ kody kähallatlarda Microsoftyn

resminamalarynda C++ dolandyrylmaýan kod diýlip hem atlandyrylýar, sebäbi CLR gurşawy onun ýerine ýetirilmegine gatnaşmaýar. C++ dolandyrylmaýan kodda programma ýerine ýetirilyän wagtynda ýady bölüp bermek we boşatmak barada özbaşdak alada etmeli bolarsyryz, siz CLR gurşawynyň içinde düzülen howpsuzlygy özbaşdak üpjün etmeli bolarsynyz. Biz C++ dolandyrylmaýan kody "öz" kody diýip hem atlandyryrs, çünki ol "öz" maşyn koduna gönüden-göni kompilirlenýär.

Goşundy bölekleyin C++ dolandyrylýan koddan, bölekleyin «öz» kodundan ybarat bolup biler, ýagny, siz ol ýa-da beýleki gurşawa bagly bolmaýarsynyz. Elbetde, dürli görnüşli kodlary garyşdyrmak bilen siz nämedir bir zatlary yitiryärsiňiz, sonuň üçin hem dine zerur bolan ýagdaýynda, mysal üçin, C++ dilindäki bar bolan «öz» goşundysyny CLR gurşawynyň dolandyryşyna geçirmekçi bolan halatynyňda şeýle edip bilersiniz. Siz, gürrüňsiz, «öz» kodunda C++ dolandyrylýan koda mahsus bolan artykmaçlyk alarsyňyz, munun üstesine hem programmanyň dolandyrylýan we dolandyrylmaýan komponentleriniň özara hereketi belli bir derejede goşmaça çykadjylaryň çekilmegine getirer. Emma dolandyrylýan kody işlap düzmek we ony giňeltmek gerek bolan mahalynda dolandyrylýan we dolandyrylmaýan kodlary garyşdyrmak mümkinçiligi örän peýdaly bolup biler, ýöne şunda kabir ýerlerde CLR gurşawynyň artykmaçlyklaryndan peýdalanmaly bolar. Elbetde, taze goşundylar işlenip düzülende onuň kodunyň dolandyrylýan ýa-da dolandyrylmaýan kod bolmalydygy ilki başda çözülmelidir.

2.4. Windows üçin C++ dilinde programmirlemek

Windows amal ulgamy esasynda ýerine ýetirilýan interaktiw goşundylar bilen iki aspekt hemişe bagly bolýar: peýdalanyjynyň onun bilen özara baglanyşykda hereket edýän peýdalanyjynyň grafiki interfeýsini (GUI) döretmek üçin kod zerurdyr, onsoham bu özara hereketi işlemek we goşundynyň peýdaly funksional mümkinçiliklerini durmuşa geçirmek üçin kod gerekdir. Visual C++ 2010 işlap düzüş gurşawy Windows amal ulgamy üçin goşundylaryň iki aspektini hem işlap düzmekde size oran gowy goldaw berýär. Koduň bir setirini hem özbaşdak ýazman GUI interfeýsli işleýän Windows programmasyny döredip bolar, siz ony şu bapda sonra görüp bilersiniz. GUI döretmek boýunca ahli binýatlyk kod Visual C++ 2010 gurşawy tarapyndan awtomatiki usulda döredilip bilner, emma awtomatiki usulda döredilýän bu kodun nahili işleýändigini unutmaly daldır, sebäbi, siz ony zerur işi ýerine ýetirmäge mejbur etmek üçin giňeltmeli we modifisirlmeli bolarsynyz, munun üçin bolsa C++ diline ahlitaraplaýyn düşünmek gerek bolar.

Şonuh üçin hem biz, Windows üçin programmirlemäge gümra bolman, ilki başda C++ diline («öz» C++) we onun C++/CLI wersiýasyna serederis. Dine C++ dilinde özüňizi ynamly duýanyhyzdan son, siz «öz» C++ we C++/CLI diliniň kömegi bilen doly manysyndaky Windows goşundylaryny işlap düzmegi öwrenersiniz. Bu bolsa C++ dilini öwrenmek arkaly sizin konsol girişini we çykyşyny peýdalanýan programmalar bilen işlejekdifizi aňladýar. Şu ýönekeý serişdeler bilen girişi we çykyşy çaklendirmegiň hasabyna siz, GUI gurmağyn we dolandyрмагyn getirýan gutulgysyz çylşyrymlylygyny örän peseltmek bilen, C++ diliniň özüniň aýratynlygynda ünsi jemlap bilersiniz. C++

dilini özleşdirmek bilen, siz doly manysyndaky Windows goşundylaryny işlap düzmekde bu dili ulanmaga ýenil we tebigy ýagdaýda geçip bilersiniz.

Visual C++ 2010 işlap düzüş gurşawy C++ dilinin iki dürli standart bilen kesgitlenen iki wersiýasyny hem doly derejede goldaýar.

C++/CLI standartynyň ISO/IEC C++ standartynyň gineldilmegi bolup durýandygy sebapli, her bir babyň birinji bölegi ISO/IEC C++ diliniň elementlerinden, ikinji bölegi bolsa C++/CLI goşmaça serişdelerinden ybarat bolýar.

C++/CLI dilinde programmalaryň ýazylmagy .NET Framework gurşawynyň artykmaçlyklaryndan doly derejede peýdalanmaga mümkinçilik berer, olar (artykmaçlyklar) ISO/IEC standartynyň C++ dilinde ýazylan programmalar üçin köplenç halatda elýeter däl. C++/CLI standartynyň C++ diliniň ISO/IEC standartynyň gineldilmegi bolup durýandygyna garamazdan, programmanyň doly derejede CLR gurşawynyň dolandyryşy astynda ýerine ýetirilmegi üçin ol CLR gurşawynyň belli bir talaplaryna laýyk gelmelidir. Bu ISO/IEC C++ standartynyň kabir serişdeleriniň CLR programmaarynda ulanylmaly daldigini ahladyr. Ýokarda aýdylanlardan aňyşyňyz ýaly, şunun ýaly çaklendirmelerin bir mysaly hem ýady bölmek we arassalamak üçin ISO/IEC C++ standartynyň serişdelerinden peýdalanmagy gadagan etmek bolup biler. Olar C++/CLI standarty bilen bir ýere sygyşmaýar we olaryň ýerine siz CLR gurşawynyň ýady dolandyryş mehanizminden peýdalanmaly, diýmek, «öz» C++ klaslary bilen däl-de, C++/CLI klaslary bilen işlemeli bolarsyhyz.

2.5. C++ standarty

ISO/IEC standarty Standartlaşdyrmak boýunca halkara guramasy (International Standards Organization - ISO) tarapyndan çap edilen ISO/IEC 14882:1998 resminamada kesgitlenildi. ISO/IEC C++ standarty C++ diliniň durnukly wersiýasyny beýan edýar, ol 1998-nji yyldan bäri dowam edip gelýar hem-de apparat kompýuter platformalarynyň we amal ulgamlarynyň köpüsi tarapyndan goldanylýar. C++ dili üçin täze standartyň taslamasy bar, ol ýakyn wagtda makullanylar diýlip garaşylýar. Visual C++ 2010 gurşawy diliň şu täze standart tarapyndan berilýän täze mümkinçilikleriniň käbirini goldaýar, olar şu kitapda beýan edilýar.

C++ dilinde ýazylan programmalar, olar tarapyndan peýdalanylýan kitaphana funksiýalarynyň, hususan-da, grafiki interfeýsi gurmak bilen bagly funksiýalaryň şunuň ýaly geçirmaniň näderejede ýenildigini ýa-da agyrdygyny kesgitleýan esasy faktor bolup durýandygyna garamazdan, bir platformadan beýleki platforma belli bir derejede ýenil geçirilip bilner. C++ ISO/IEC standarty programmalary hünarli işlap düzüj ileriň saýlap alýan esasy guralydyr, sebäbi, ol giňden goldanylýar we şu günki günde programmirlemegiň has kuwwatly elýeter dilleriniň biri bolup durýar.

C++/CLI dili - bu C++ diliniň wersiýasy bolup, ol ECMA-355 standartynda kesgitlenen diliň umumy infrastrukturasy (Common Language Infrastructure -CLI) has gowy goldamak üçin C++ ISO/IEC standartyny giheldýär. Bu standartyň ilkinji nusgalary 2003-nji yylda peýda boldy we .NET Framework gurşawynda C++ programmalaryny goldamak üçin Microsoft korporasiýasy tarapyndan berlen tehnikä spesifikasiýalaryň esasynda işlenip düzüldi. Başga sözler bilen aýdylanda, umuman CLI infrastrukturasy hökmünde, aýratynlykda C++/CLI dili hökmünde Microsoft

korporasiýasynda döredi we .NET Framework gurşawyny goldamak üçin niýetlenendir. Elbetde, CLI infrastrukturasynyň we C++/CLI dilinin standartlaşdyrylmagy Windows amal ulgamyndan tapawutlanýan gurşawlarda ornaşdyrmalaryň peýda bolmak ahtimallygyny ep-esli ýokarlandyrdy. C++/CLI dilinin C++ ISO/IEC standartynyň giňelmegi bolup durýandygyna garamazdan, C++ ISO/IEC standartynyň CLR gurşawynyň dolandyryşynda dolulygyna ýerine ýetirmek üçin niýetlenen programmalar peýdalanylmalý däl serişdelerinin bardygyna baha bermek möhümdir. Sonra siz bu barada ginişleýin bilersiniz.

CLR gurşawy «öz» gurşawynyň önünde düýpli artykmaçlyklary göz önünde tutýar. C++/programmasynyň CLR gurşawyna gözükdirilmegi olaryn howpsuzlygyny ýokarlandyrýar we ISO/IEC C++ standartynyň ahli serişdeleri peýdalanylanda mümkin bolan potensial ýalňyşlar babatynda gowşaklygy peseldýär. Şeýle hem CLR gurşawy ýokara derejeli programmirlеме dillerinin arasyndaky laýyk gelmezligi aradan aýyrýar, bu maksatlaýyn gurşawy standartlaşdyrmagyň hasabyna gazanylýar, munun üçin kompilyasiýa ýerine ýetirilýär, bu bolsa C++ dilinde ýazylan modullary C# ýa-da Visual Basic, ýaly beýleki dillerde ýazylan modullar bilen utgaşdyrmaga mümkinçilik berýar.

Atributlar (nyşanlar)

Atributlar - bu C++/CLI dilinde programmirlемегіň öňdebaryjy serişdesidir, ol sizin koduňyza beýan ediji bildirişleri goşýar. In yönekey derejede atributlary programmanyň belli bir elementlerini annotirlemek üçin ulanyp bolar, yöne atributlar diňe beýan ediji maglumatlar bolmak bilen çaklenmeyar. Atributlar kody kompilyasiýa etmegin usulynyň üytgemeginiň hasabyna ýa-da goşmaça mümkinçilikleri goldaýan kody döretmegiň hasabyna ýerine ýetirilýän wagtynda kodun özüni alyp barşyna täsir edip biler. C++/CLI dili üçin atributlaryň öran köpüsi elýeterdir, mundan başga-da, öz atributlaryňy hem döredip bolýar.

Atributlara ginişleýin seretmek şu kitabyň çagine girmeyar, yöne olar bu kitabyň bir ýa-da iki ýerinde peýdalanyljakdygy üçin ýatlanylýar, ondan siz obýektleri faýla nähili ýazmalydygyny bilersiniz.

Konsol goşundylary

Windows üçin goşundylary işlap düzmek bilen bir hatarda Visual C++2010 gurşawy C++ programmalaryny ýazmaga, kompilirlemäge we barlamaga hem mümkinçilik berýar, olar Windows üçin zerur bolan ahli zady, ýagny konsol programmalaryny (buýruk setiri programmalary) özünde jemleýar. Visual C++2010 işlap düzüş gurşawynda bu programmalar konsol goşundylary diýlip atlandyrylýar, sebäbi siz olar bilen simwol düzgüninde işleýan klawiaturanyň we ekranyň kömegi arkaly özara hereket edýarsihiz.

Konsol goşundylaryny ýazmak Windowsy programmirlемегіn esasy maksadyndan gýsarma ýaly bolup görünmegi mümkindir, yöne gürrüniň C++ dilini öwrenmek (bu Windows amal ulgamy üçin programmirlемäge girişmezden ön oran zerurdyr) barada gidýan wagtynda bu in gowy usuldyr. In yönekey Windows programnasynda hem öran köp kod bolýar, bu Windowsa laýyk programmirlemek bilen bagly kynçylyklaryň sizin üçin başlangyç C++ dilinin ähmiýetini peseltmezligi öran möhümdir. Şonuh üçin hem kitabyň C++ dilinin esaslarynda jemlenen başky baplarynda biz Windows dünýäsiniň „agyr“ koduna geçmezden ön, birnaçe „ýenil“ konsol goşundylaryna seretmäge biraz wagtymyzy sarp ederis.

Şeýlelikde, C++ dilini öwrenmek bilen, siz işleýän gurşawynyň barada alada etmän, dil meselelerinde ünsi jemläp bilersiňiz. Öwrenmegiň barşynda özüňizni işlap işlap düzýän konsol goşundylaryňyzda siz dine tekstli interfeýs bilen iş salyşarsyňyz, ýöne bu C++ diliniň esaslaryna düşünmek üçin ýeterlikdir, sebabi diliň kesgitlemesinde haýsydyr bir grafiki interfeýs serişdeleriniň beýany ýokdur. Elbetde, grafiki peýdalanyjy interfeýsini programmirlemek baradaky zerur bolan ähli maglumatlar «öz» C++ dilinde MFC kitaphanasyny ulanmak bilen, şeýle hem CLR gurşawyny peýdalanyň, Windows Formsy ulanmak bilen Windowsa laýyk programma ýazmaga nobat ýetende berler.

Konsol goşundylarynyň iki görnüşü bar, siz olaryň ikisi bilen hem işlemeli bolarsyňyz.

Windows üçin programmirlemegiň konsepsiýasy

Windows amal ulgamy üçin programmirlemäge biziň çemeleşmämiz Visual C++2010 gurşawy tarapyndan berlen gurallaryň doly peýdalanylmagyna esaslanýar. Taslamany düzmegiň Visual C++2010 gurşawynyň içinde gurlan serişdeleri ýönekeý Windows programmaýaryny hem goşmak bilen, dürli amaly programmaýaryň giň topary üçin kodun esasy awtomatiki usulda döretmäge mümkinçilik berýär. CLR gurşawy üçin işlenip taýýarlanylýan Windows programmalary üçin siz ýene-de awtomatiki usulda döredilýän köp kody alýarsyňyz. Siz Windows Formsy peýdalanmak bilen diňe uly bolmadyk goşmaça kody ýazmagy talap edýän, kähalatda bolsa ony hem talap etmeýän gutarnykly goşundylary döredip bilersiniz. Taslamany döretmek - bu Visual C++2010 gurşawynda ahlil goşundylar we komponentler işlenip düzülen mahalynda başlangyç nokatdyr we munuň nähili edilýändigini barada düşünje almak üçin şu babda siz Windows üçin programmalaryň ilkinji nusgasyny goşup, kabir mysallary döretmek işi bilen tanyşyp bilersiniz.

Windows programmalarynyň adaty konsol programmalaryndan tapawutlanýan gurluşy bardyr, olar buýruk setirinden işe girizilýär, üstesine-de olar örän çylşyrymlydyr. Konsol programmasynda siz klawiraturadan giriş alyp bilersiniz, gönüden göni buýruk setirine çykyş iberip bilersiniz, Windows programmasy bolsa kompýuteriň giriş-çykyş serişdeleri bilen diňe Windows amal ulgamy tarapyndan berlen funksiýalaryň: üstünden işläp bilýär; aparat serişdelerinin gönüden göni elýeterli bolmagyna ýol berilmeýär. Windows amal ulgamynda birwagtda birnäçe programmalaryň işjeň bolup biljekdigi sebapli, Windows amal ulgamy olaryň haýsy biriniň giriş almadygyny (syçanjygyň düwmesine basmak ýa-da klawiaturada klawişi basmak) kesgitleýär we degişlilikde, gerek programma habar berýär. Başga sözler bilen aýdylanda, peýdalanyjy bilen özara hereketleriň ählisini umumy dolandyrmak Windows amal ulgamy tarapyndan amala aşyrylýar.

Munun üstesine-de, peýdalanyjy bilen Windows goşundysynyň arasyndaky interfeýsiň tebigatyna göre wagtyň islendik pursatynda programma köp dürli täsir edip bolýar. Peýdalanyjy menýudaky islendik nokady saýlap alyp, gurallar paneliniň düwmesine basyp biler ýa-da goşundylar penjiresiniň islendik ýerine basyp biler. Windowsyň gowy taslanan goşundysy wagtyň islendik pursatynda islendik görnüşdäki girişi işlemäge taýýar bolmalydyr, sebäbi nahili tasiriň boljakdygy öňünden mälim bolmaýar. Peýdalanyjynyň şu hereketleriniň ählisi amal ulgamynyň birinji instansiýasynda kabul edilýär we şol instansiýa tarapyndan oňa waka (event) hökmünde

seredilýar. Goşundynyň peýdalanyjy interfeýsi tarapyndan kabul edilen waka adatça programmanyň kodunyň belli bir fragmentiniň ýerine ýetirilmegine getirýar. Şeýlelikde, programmanyň näme edýandigi peýdalanyjynyň hereketiniň yzygiderliligi bilen kesgitlenýär. Şu görnüşde işleýän programmalar wakalar tarapyndan dolandyrylýan programmalar (event-driven programs) diýlip atlandyrylýar we ýerine ýetirmegin ýeke-täk yzygiderliligi bolan adaty proseduraly programmalardan tapawutlanýar. Proseduraly programma maglumatlary girizmek programmanyň öz kody tarapyndan dolandyrylýar we diňe programmanyň ony kabul eden halatynda ýerine ýetirilip bilner; munuň tersine, Windows programmasy, öňi bilen, peýdalanyjynyň ýa-da Windows amal ulgamynyň özüniň hereketleri sebäpli yüze çykan wakalara seslenýan kodun fragmentlerinden durýar. Programma kodun özara bagly bolmadyk bloklarynyň köpdügi sebäpli, güýçli derejede fragmentlere bölünen ýaly bolup görünüp biler, ýöne ony bir bitewilige baglaýan esasy faktor -bu Windows amal ulgamynyň özüdir. Siz öz programmanyzy hereketleriň belli bir toplumyny ýerine ýetirmek üçin Windowsy indiwiđuallaýyk getirme hasap edip bilersiniz.

Elbetde, menýudan nokady saýlap almak ýa-da syçanjygyn düwmesine basmak ýaly daşarky dürli wakalara hyzmat edýan modullaryň ählisi üçin anyk programmadaky maglumatlaryň goşundylaryna mahsus bolan umumy toplum elýeterdir. Bu goşundylar programmanyň niýetlenen ugruna degişli bolan maglumatlary, mysal üçin, redaktorda tekstiň blogyny ýa-da beýsbol toparynyň oynunyň netijelerine syn etmek üçin niýetlenen programmada oýunçylaryň rekordlar tablisasyny, şeýle hem programmanyň ýerine ýetirilýan wagtynda bolup geçen beýleki käbir wakalar baradaky maglumatlary öz içine alýar. Maglumatlaryň bu umumy toplumy programmanyň garaşsyz bolup görüňän dürli böleklerine özara ylalaşykly hereket etmäge mümkinçilik berýär. Soňra biz muha has gihişleýin serederis.

In bir yönekey Windows programmasy hem koduň birnaçe setirlerini öz içine alýar, gürrüň Visual C++2010 işläp düzüş gurşawynyň ussalary tarapyndan awtomatiki usulda döredilen Windows programmalary barada gidýan wagtynda bolsa, „birnaçe” setir „köp” setire öwrülýar. C++ dilinin işleýşine düşünmek prosesini ýenilleşdirmek üçin mümkin boldugyça yönekeyleşdirilen kontekst gerekdir. Gowy ýeri Visual C++2010 gurşawy şunuň ýaly mümkinçiligi üpjün edýar.

Işlap düzmegin integrirlenen gurşawy name?

Visual C++2010 işlap düzüş gurşawy bilen bilelikde iberilýan *işlap düzmegin integrirlenen gurşawy* (Integrated Development Environment-IDE) - bu C++ dilinde programmalary döretmek, kompilýasiýa etmek, düzmek we barlamak üçin niýetlenen, doly öz-özüni üpjün edýan gurşawdyr. Şeýle hem ol C++ dili boýunca öran gowy okuw gollanmasy bolup durýar (aýratyn hem gowy kitap bilen bilelikde).

Visual C++ 2010 gurşawy C++ dilinde programmalary ýazmagy ýenilleşdirmek üçin niýetlenen, doly integrirlenen gurallaryň köplüginu öz içine alýar. Olaryn kabiri bilen siz şu bapda tanyşarsynyz, yöne serişdeleriň we mümkinçilikleriň tukat abstrakt sanalyp geçilmeginiň ýerine ilkibaşda IDE işi harada umumy düşünje berler, soňra bolsa, öwrenilýan materialyň kontekstinde jikme-jiklikler tapgyrlaýyn beýan ediler.

Visual C++2010 işlap düzüş gurşawynyň IDE bölekleri hökmünde iberilýan fundamental düzümler bölekleriniň sanawy redaktory, kompilýatory, düzüjini we kitaphanany öz içine alýar. Bular C++ dilinde programmalary ýazmak we ýerine

ýetirmek üçin zerur bolan esasy gurallardyr. Olaryn bellenen maksatlary aşakda beýan edilýär.

Redaktor

Redaktor interaktiw gurşawdan ybaratdyr, onda siz C++ diliniň başlangyç koduny döredip we redaktirlap bilersiniz. Size tanyş bolan fragmentleri kesip aýyrmak we goymak ýaly adaty serişdeler bilen bir hatarda redaktor dilin dürli elementleriniň reňkli görkezilmegini hem üpjün edýär. Redaktor C++ diliniň esasy konstruksiýalaryny awtomatiki usulda tanaýar we olara ähmiýetine laýyklykda reňk berýär. Bu bolsa diňe kody okamagy ýenilleşdirmän, eýsem esasy sözler girizilen mahalynda ýalňyşlary hem açyk görkezýär.

Kompilýator

Kompilýator başlangyç kody adaty koda öwürýar, ýalňyşlary yüze çykarýar we kompilýasiýa etmegiň barşynda olar harada habar berýär. Kompilýator nadogry hem tanap bolmaýan programma kody bilen bagly ýalňyşlaryň gini toparyny, şeýle hem mysal üçin, programmanyň hiç wagty ýerine ýetirilmejek gurluş ýalňyşlyklaryny yüze çykarýar. Kompilýator tarapyndan döredilen obýektiw kod obýektiw faýllar diýilýänlere ýerleşdirilýär. Kompilýator tarapyndan döredilýän obýektiw kodyň iki görnüşi bardyr. Obýektiw kodly faýllaryň adaty giňeltmeleri bardyr .obj.

Düzüji

Düzüji başlangyç koduň faýllaryndan kompilýator tarapyndan döredilen dürli modullary bir ýere jemleýär, C++ diliniň düzüminde iýerilýän kitaphalardan zerur bolan modullary goşýar we ýerine ýetirilýän bir bitewi gömüşe düzýar. Şeýle hem düzüji, eger siziň programmaňyzyň haýsydyr bir bölegi galdyrylan ýa-da ýok bolan kitaphana komponentine salgylanma yüze çykarylan bolsa, ýalňyşlary yüze çykaryp we olar barada habar berip biler.

Kitaphanalar

Kitaphanalar ilibrariýa) - bu G++ dilini goldaýan we giheldýän, deslapdan ýazylan proseduralaryň toplumydyr, ol siziň ygtyýaryňyza koduň standart, hünarli işlenip düzülen birliklerini berýar, siz olary standart, yggy gabat gelýän amallary ýerine ýetirmek üçin öz programmalaryňyza goşup bilersiniz. Visual C++ 2010 işlap düzüş gurşawynyň dürli kitaphanalaryndan bolan proseduralar tarapyndan durmuşa geçirilen amallar olary özbaşdak taýýarlamak we harlamak üçin sarp ediljek zahmeti tygşytlamagyň hasabyna siziň öndürijiligini has ýokarlandyryr. .NET Framework kitaphanasyny ýatlap geçipdik, yöne ondan başga-da kitaphanalar örän köpdür, hatda bu ýerde sanap geçerden hem oran köpdür, yöne olaryň has möhümlerini görkezeliň.

C++ standart kitaphanalar ISO/IEC C++ kompilýatorlarynyň ählisi üçin umumy bolan proseduralaryň binýatlyk toplumyny kesgitleýar. Onda san funksiýalaryny goşmak bilen, kwadrat köki hasaplamak, trigonometriki funksiýalar, simwollary we setirleri simwollaryň klassifikasiýasy ýaly işlemek we simwol setirlerini denesdirmek proseduralary ýaly köp funksiýalary içine alýan kiçi programmalaryň giň topary bardyr. Siz ISO/IEC C++ öwrenmegiň barşynda olaryň köpüsi bilen tanyş bolarsyňyz. ISO/IEC C++ standartynyň C++/CLI giheldilmegini goldaýan kitaphanalar hem bar.

«Öz» penjire goşundylary MFC (*Microsofts Foundation Classes*) diýlip atlandyrylýan kitaphana tarapyndan goldanylýar. MFC kitaphanasyny goşundynyň grafiki peýdalanyjy interfeýsini gurmak üçin zerur bolan işleri ep-esli azaltmaga mümkinçilik

berýär. C++ diliniň inçelikleri beýan edilenden soň siz MFC kitaphanasy barada köp zat bilersiniz. Grafiki interfeýsi gurmagyň serişdeleriniň toplumyny içine alýan beýleki bir kitaphana Windows Forms diylip atlandyrylýar. Ol takmynan MFC kitaphanasyna meňzeşdir, .NET Framework gurşawynda ýerine ýetirilýan penjire goşundylaryny döretmek üçin hyzmat edýar. Öz wagtynda siz goşundylar işlenip düzülende ondan nähili peýdalanylmalydygyny bilersiniz.

3. MICROSOFT VISUAL STUDIO WIZUAL PROGRAMMIRLEME GURŞAWY WE ONUN MÜMKİNÇİLİKLERİ

3.1. IDE peydalanmak

Şu kitapda beyan edilen programmalary işlap düzmek we yerine yetirmek IDE-de (integrirlenen işlap düzüş gurşawy) amala aşyrylar. Visual C++2010 işlap düzüş gurşawy işe goyberilende menzeş goşundy penjiresini görersiniz.

Penjiräniň çep bölegi çözügütleriň geçirijisi (Solution Explorer) diýlip atlandyrylyar, başky sahypany (Start page) öz içine alyan yokarky sag tarapdaky penjire - bu redaktoryň (Editör) penjiresidir, penjiräniň aşagynda görünýän goşmaça sahypa bolsa çykyş (Output) penjiresidir. Çözügütleri geçiriji penjire programmanyň faýllary boýunça hereket etmegi amala aşyrmaga, olarda bar bolan maglumatlary redaktoryň penjiresinde görkezmäge, şeýle hem siziň programmaňyza täze faýllary goşmaga mümkinçilik beryär. Çözügütleri geçiriji penjiresi bir goşmaça sahypany öz içine alýar, bu sahypa serişdeleriň görkezilişini (Resource View) görkezýär we siz View (Gömüş) menýusynda haysylaryň şöhlelendirilmelidigini görkezip bilersiniz. Redaktoryň penjiresi - bu siziň öz programmaňyzyň başlangyç kodyny we beyleki komponentlerini girizýän we modifisirleýän yeriňizdir. Çykyş penjiresi programma kompilyasiya edilende we düzülende alnan habarlary görkezýär. View (Gömüş) menýusynda siz görkezmek üçin beyleki penjireleri hem saylap bilersiniz.

Penjiräniň Visual C++ goşundysynyň penjiresindäki öz yerinde berkidilmän biljekdigine hem üns beriň. Eger penjiräni berkitmän, ony üýtgäp durýan edesiniz gelýän bolsa, onun adynyň üstüne baryp syçanjygyň sag düwmesini basyň we peýda bolan kontekst menýusynda Float (Üýtgeýän) nokadyny saýlaň. Düzgün bolşy yaly, olary bu ýerde berkidilmedik ýagdaýda görkezeliň. Onun adynyň üstüne baryp syçanjygyň sag düwmesini basmak we peýda bolan kontekst menýusynda Dock (Berkidiji) nokadyny saýlap almak bilen penjiräni berkidilen ýagdaýa getirip bilersiniz.

3.2. Gurallar paneliniň mümkinçilikleri

Gurallar paneliniň ýaýlasýnda syçanjygyň sag düwmesini basmak bilen, peýda bolan kontekst menýusynda görkezilyän gurallar panelini (toolbar) saýlap bolar. Şunda häzirki pursatda görkezilyän paneller elýeter panellerin sanawynda bellik bilen görkezilendir. Siz bu yerde haýsy gurallar panelleriniň wagtyň islendik pursatynda görnüp durmalydygy barada çözügüt kabul edýarsiniz. Siz Build (Yygnama), Class Designer (Klaslaryň konstruktory), Debug (Düzetme), Standard (Standart) we View Designer (Görkezmeler konstruktory) nokatlary bellemek bilen, gurallar toplumyny saýlap alyp belersiňiz. Gurallar paneliniň çepindäki çal yerde düwmä basylmagy baýdajyklary goýýar ýa-da eger eýýäm bar bolsa, olary aýyrýar. Bellenen paneller görünýär, bellenmedikler gizlenýär.

IDE penjiresi sizin pikir edişiniz ýaly geljekde gerek bolup biljek gurallar panelleri bilen doldurylmaly däldir. Olaryň käbiri özlerine zerurlyk yüze çykanda awtomatiki usulda peýda bolýar, şol sebapli hem panellerin standart toplumynyň has amatly bolup durandygy bilen siz ylalaşarsynyz. Goşundynyň işlenip düzülýän wagtynda belli bir döwürlerde size häzirki wagtda görünmeýän paneller gerek bolup biler. Şunda siz paneller meýdanynda syçanjygyň sag - dü.wmesine basmak we kontekst menýusynda talap edilýän paneli saýlap almak bilen, görünýän gurallar paneliniň toplumyny üytgedip bilersiniz.

Bellik. Windowsyň köp sanly beýleki goşundylary ýaly, Visual C++2010 işlap düzüş gurşawynyň gurallar panelinde hem yüze çykýan gönükdirijiler bardyr. Olary görmek üçin syçanjygyň kursoryny degişli düwmä getirip, bir ýa-da iki sekund garaşyň, siz bu düwmaniň funksiýalarynyň gysgaça beýany bolan ak belgilini görersiniz.

Gurallaryň berkidilýän panelleri (dockable toolbar) - bu syçanjygyň kömegi bilen penjiredäki islendik yere geçirip bolýan paneldir. Siz gurallar paneliniň islendigini goşundy penjireleriniň dört tarapyňyň islendik birinde berkidip bilersiniz. Eger gurallar paneliniň ýaýlasynda syçanjygyň sag düwmesine basyp, peýda bolan kontekst menýusynda Customize (Sazlamak) nokadyny saýlap alsaňyz, onda Customize (Sazlamak) dialog penjiresi açylar. Siz bu yerde açylýan Modify Selection (Saýlanyp alnany üytgetmek) menýuda degişli nokady saýlap almak bilen, gurallaryň belli bir paneliniň berkidilmeli yerini görkezip bilersiniz. Peýdalanyjy tarapyndan gurallar paneli saýlanyp alnandan son çepde Build we sagda düwme basylanda Selection (Saýlanyp alnany üytgetmek) görkezilyär.

Gurallar panelleriniň Visual C++2010 işlap düzüş gurşawynda peýdalanylýan piktogrammalarynyň köpüsi size Windowsyň beýleki goşundylary boýunca tanyşdyr, ýöne siz Visual C++2010 kontekstinde bu piktogrammalaryň name edýändigini bilýän dälseňiz, sonuň üçin hem olary peýdalanylyşyna göre beýan edeliň.

Resminamalar

Visual C++2010 işlap düzüş gurşawy, şeýle hem onun serişdeleri we mümkinçilikleri barada gihişle yin maglumatlaryň gerek bolýan wagtlyary seýrek däldir. Önümin resminamasynyň elyeter bolmagy üçin Ctrl+Alt+F1 klawişlerini utgaşdyryp basyň. Help (Maglumat) menýusy resminamalar boýunca dürli salgylanmalary, kodlaryň mysallaryna we tehniği goldawa elyeterlilikigi öz içine alýar.

Taslamalar we çözümler

Taslama (project) - bu ony emele getirýän belli bir görnüşdäki programmalaryň ählisi üçin konteyner bolup durýar. Bu konsol, penjire programmasy ýa-da bir ýa-da birnäçe başlangyç faýllardan durýan, bir ýa-da birnäçe kömekçi faýllaryň koduny öz içine alýan başga tipli programma bolup biler. Taslamanyň faýllarynyň ählisi taslama bukjasynda, taslama baradaky ginişleýin maglumat bolsa - XMI faýlynda saklanýar, bu bukjada .vcproj ginelmesi hem bolýar. Taslama bukjasynda taslamany kompilyasiýa etmegiň we düzmegiň çykyş faýllaryny saklamak üçin peýdalanylýan beýleki bukjalar hem bardyr.

Çözgüdin (solution) ideýasy onun adynda berilyär. Çözgüt - bu ähli programmalary we maglumatlary işlap geçmek bilen bagly belli bir problemanyň

çözümden ybarat bolan beýleki serişdeler (resurslar) bilen birleşdirmek üçin mehanizmidir. Meselem, kabir telekeçilik amallary üçin sargytlary girizmegiň bölünen ulgamy dürli-dürli birnäçe programmalaradan ybarat bolup biler, bu programmalaryň her biri bir çözümdäki taslama görnüşinde berlip bilner. Şeýlelikde, çözüň - bu bir ýa-da bimaçe taslamalar baradaky ähli maglumatlaryň toplanan bukjasydyr, şunda taslamalaryň bukjalary çözüň bukjasyna salnandyr. Çözüdiň taslamalary baradaky maglumat .slp we .suo ginelmeleri bolan iki faýlda saklanýar. Siziň taslamany döredýän wagtyňyzda täze çözüň, eger siz taslamany bar bolan çözüde goşmadyk bolsanyz, awtomatiki usulda döredilýär.

Eger siz taslamany çözüň bilen bilelikde döredýän bolsaňyz, onda soňra şol çözüde goşmaça taslamalary goşup bilersiňiz, islendik görnüşdäki taslamalary bar bolan çözüde goşup bilersiniz, ýöne adaty bu şol çözüdiň eýýäm bar bolan taslamalary bilen haýsy hem bolsa bir görnüşde bagly bolan taslamalar bolar. Adaty, eger başgaça hereket etmek üçin degerli sebäp yok bolsa, her bir taslama aýratyn çözüde degişli bolmalydyr. Şu kitaba goşulan ähli mysallar olarda öz çözüňleri bolan taslamalardan ybaratdyr.

Taslamanyň kesgitlemesi

Visual C++2010 gurşawynda programma ýazylanda ilkinji ädim taslamany döretmek bolup durýar. Munuh üçin baş menýuda File-New-Project (faýl-Taze-Taslama) nokadyny saýlap almak ýa-da Ctrl+Shift+N klawişleri utgaşdyryp basmak ýeterlikdir. Şeýle hem New Project (Taze taslama) elementinden başlangyç sahypany açyp bolar. Kody we siziň programmaňyzy emele getirýän beýleki ähli maglumatlary kesgitleýän faýllaryň sanawy bilen bir hatarda taslamanyň bukjasyndaky taslamanyň XMI faýly hem Visual C++2010 işlap düzüş gurşawynyň peýdalanylýan parametrleri saklaýar.

Konsol goşundysynyň mysaly

Win 32 konsol goşundysynyň taslamasyny döretmek

Häzir biz konsol goşundysynyň taslamasyny döretmäge synanyşarys. ilki bilen menýuda File-New-Project (Faýl-Taze-Taslama) nokadyny saýlap alyň ýa-da New Project (Taze taslama) dialog penjiresini açmak üçin ozal sanalyp geçilen beýleki mümkinçiliklerin birinden peýdalanyň.

Bu dialog penjiresiniň çep panelinde taslamalaryň tipleri görkezilendir, olary döredip bolýar. Şu ýagdaýda Win 32 piktogramma basyň. Şeýle hem bu goşundylary döredýän, ussa (master) habar berer, ol başlangyç maglumatlary bolan taslamany döreder. Birinji panel çepde saýlanyp alnan taslamalaryň tipi üçin elýeter bolan şablonlaryň sanawyny görkezýar. Saýlanyp alnan şablon goşundylaryň ussasy tarapyndan taslamany emele getirýän faýllar döredilende peýdalanylýar. Indiki dialog penjiresinde size şu dialog penjiresiniň OK düwmesi basylandan son döredilen faýllary sazlamak mümkinç iligi berler. Mysaly şablonlaryň köpüsi üçin programmanyň başlangyç modullarynyň binýatlyk toplumu awtomatiki usulda döredilýär.

Şundan son Name: (Ady) redaktirlenýän meýdanda taslamanyň degişli adyny ýazyn. Ony mysal üçin, Exl-01 diyip atlandyryň ýa-da öz garaýşyňyza göre taslama başga at saýlap alyň. Visual C++2010 işlap düzüş gurşawy faýllaryň uzyn atlaryny hem goldaýar, sonun üçin saýlap almak mümkinçiligi hiç hili çäklendirilen däldir. Çözüň bukjasynyň ady redaktirlenýän meýdanyň aşagynda peýda bolýar we duýdurmazdan taslamanyň ady bilen laýyk gelýar. Işlegiňize göre ony üýtgedip bilersiňiz. Dialog penjiresi siziň taslamaňyzy öz içine alýan çözüdiň

ýerini üytgetmäge hem mümkinçilik berýär, munun üçin Lokation: (yerleşýan yeri) redaktirlenýän meýdan niyetlenendir. Eger siz dine taslamanyň adyny girizseňiz, onda çözgüt bukjasy Lokation: meýdanynda görkezilen ýolda awtomatiki usulda şu at bilen bukja giriziler. Çözgüt bukjasy, eger ol bar bolsa, duýdurmazdan döredilýär. Eger siz çözgüt bukjasy üçin başga yol görkezmekçi bolsahyz, onda onun adyny Lokation: meýdana giriziň. Çözgüdi yerleşdirmegiň başga ýoluny saýlap almak üçin Browse (Syn) düwmesinden peýdalanyň bilersiňiz. OK düwmesine basylanda Application Wizard ussasyňyň (Goşundylary döredýan ussa) dialog penjiresi açylar.

Dialog penjiresinde gündelik işeň enjamlar görkezilýär. Eger siz Finish (Taýýar) düwmesine bassanyz ussa şolaryň esasynda taslamanyň faýllaryny döredýär. Şu ýagdaýda goşundylary sazlamagyň sahypasyny açmak üçin, çep bölekde Application Settings (Goşundylary sazlamak) salgysyna basyp bilersiniz.

Application Settings (Goşundylary sazlamak) sahypasy taslama babatynda ulanylmaly parametrleri saýlap almaga mümkinçilik berýar. Köp halatlarda C++ dilini öwtrenmegiň barşynda taslamalar döredilende siz Empty project (Boş taslama) baýdajygyny goýup bilersiniz, yöne häzirlikçe ähli zady önküligine goýuň we Finish düwmesine basyň. Şondan son goşundylar ussasy duýdurmazdan ähli faýllary bolan taslamany döreder.

Taslama bukjasy sizin taslamanyň ady hökmünde görkezen adynyzy alar we taslamanyň kesgitlemesini emele getirýän ähli faýllary öz içinde saklar. Eger siz ony üytgetmeseňiz, onda çözgüt bukjasy hem taslama bukjasynyň adyny alar we çözgütleri beýan edýan faýllar bilen bilelikde taslama bukjasyny öz içinde saklar. Eger siz çözgüt bukjasynyň içindäkilere seretmek üçin Windows geçirijisinden peýdalanyň bolsanyz, onda aşakda getirilýan dört faýly görersiniz:

- Çözgüde girýan taslamalar baradaky maglumatlary öz içine alýan, .sin ginelýan faýl.
- Çözgüt üçin peýdalanyjy tarapyndan saýlanyp alnan parametrleri öz içine alýan, .suo ginelýan faýl.
- Çözgüt üçin *Intelisense* maglumatlary öz içine alýan, .sdf ginelýan faýl. *Intellisense* ulgamy - bu awtomatiki usulda tamamlanmagy we başlangyç kodun konstruksiýalaryny redaktör penjiresine çykarmagy üpjün edýan serişdedir.
- Taslamanyň ýagdaýy baradaky baradaky maglumatlary öz içine alýan, .opensdf gineldilmeli faýl. Bu faýl taslamanyň açyk wagtynda bolýar.

Eger siz taslama bukjasyna seretmek üçin Windows geçirilişinden peýdalanyň bolsanyz, onda ilki başda taslamanyň faýllarynyň içindäkileriň beýany bolan ReadMe.txt faýlyny goşmak bilen, yedi faýly görersiňiz. Siziň döreden taslamanyz Visual C++2010 işlap düzüş gurşawynda awtomatiki usulda açylar we çepdäki panelde görkezilen görnüşi alar.

4. MAGLUMATLARYŇ ESASY GÖMÜŞLERİ, OLARYŇ ÜSTÜNDE ÝERİNE ÝETIRILÝAN AMALLAR

Şu bapda biz C++ dilinden programmirlmegiň esaslaryna ýuzleneris. Ony ahyryna çenli okap çykanyňyzda siz döp bolan görnüşde, ýagny, giriş-işlap geçmek-çykyş görnüşinde C++ dilindäki ýonekeý programmalary ýazyp bilersiňiz. Öňki bapda aydylyşy ýaly, biz sizi ISO/IEC C++ standart diliniň serişdeleri bilen tanyşdyrayys, sonra bolsa ISO/IEC C++/CLI diliniň tapawutly we goşmaça iähtlçrini açyp görkezeris.

Işleyan programmalaryň esasynda dilin esaslaryny öwrenmek bilen, siz Visual C++2010 işlap düzüş gurşawyny ulanmakda tejribe toplamak mümkinçiligini alarsyňyz. Bu mysallaryň her biri üçin ony gurmazdan we işe goýbermezden öň taslama döretmeli bolarsyňyz.

4.1. C++ dilinde programmanyň gurluş düzümi

Visual C++2010 işlap düzüş gurşawynda konsol goşundylary ýaly işe goýberilýän programmlar buýruk setirindäki maglumatlary okaýarlar we netijeleri hem şol ýere çykarýarlar. Olaryň işleyşine düşünmek üçin bilimiň ýeterlik möçberiniň yok wagtynda goşundylar penjiresini döretmek we dolandyrmak bilen bagly çylşyrymlylyklara çuňlaşmak zerurlygyndan gaça durmak üçin, C++ dilini ilki başda öwrenmegiň barşynda siziň ýazjak ähli programma mysallaryňyz konsol programmalary, ýagny, Win32 konsol programmalary, şeýle hem .NET konsol programmalary bolar. Bu size ünsüňizi dolulylygyna C++ dilinde jemlemäge mümkinçilik berer. Dine ony ýeterlik derejede özleşdirenizden soň, siz penjire goşundylaryny döretmäge we olary dolandyrmaga taýýar bolarsyňyz. Ilki başda siz sol programmalarynyň nähili gurlandygyny bilersiniz.

C++ dilindäki programma bir ýa-da ondan köp funksiýadan durýar. !ISO IEC standartynyň C++ düindäki her bir programmasynda main() funksiýasy bolýar, islendik ölçegdäki C++ programmalarynyň ählisi birnäçe funksiýalardan - programmanyň ýerine ýetirilip baslanýan main() funksiýasyndan we beýleki funksiýalaryň birnäçesinden durýar.

Funksiýa - munuň özi özboluşly ady bolan koduň öz-özünü üpjün edýan blogy bolmak bilen, ol ony ýerine ýetirmek maksady bilen işe goýbermek üçin peýdalanylýar. 1-nji bapda görkezilişi ýaly, goşundylar ussasy (*Application Wizard*) tarapyndan döredilen Win 32 konsol programmasynyň tmain diýlip atlandyrylýan baş funksiýasy bardyr. Hereket edýan ylalaşyk boyunca Visual C++ gurşawynda programmada Unicode simwolynyň peýdalanylýandygyna ýa-da dalanyymaýandygyna baglylykda main ýa-da wmain diýlip atlandyrylmalydyr. Microsoft üçin main we _tmain ady mahsusdyr. C++ diliniň ISO/IEC standartyna laýyk gelýan baş funksiýanyň ady main atlandyrylýar. Men main adyny ISO/IEC diliniň ähli mysallarynda peýdalanaryn, sebäbi bu geçirip bolýan mümkinçilikdir.

Eger siz Microsoft Visual C++ işlap düzüş gurşawynda kody kompilirlemegi maksat edinýan bolsaňyz, onda Microsoft üçin mahsus bolan main adyny peýdalanyň bilersiniz. Application Wizard goşundylar ussasyňy peýdalanmak üçin main telefunksiýasynda berlen kódy `_tmain` funksiýasyna nusga almak ýeterlikdir.

Buýruk setiriniň tipiki programmasy 4.1 suratda görkezilişi ýaly struktirlenip bilner. 4.1 suratda tipiki konsol programmasyny ýerine ýetirmegiň `main()` funksiýasyndan baslanýandygy görünyar. Dolandyryş `main()` funksiýasyndan `input_names ()` funksiýasyna berilýar, ol onuň `main()` funksiýasyndan çagyrylan nokadynyň yzyýanyndaky orná gaýtarýar. Soňra `main()` funksiýasyndan `sort_names ()` funksiýasy çagyrylýar we dolandyryş `main()` funksiýasyna gaýtarylandan soň `output_names ()` final funksiýa çagyrylýar. Netiejede, çykarmak tamamlanandan soň dolandyryş `main()` funksiýasyna gaýtarylýar.

Elbetde, dürli programmalaryň düybünden dürli-dürli funksional gurluşlary bolup biler, emma olaryň ählisi ýerine ýetirmegi ilki `main()` funksiýasyndan baslaýarlar. Programmany funksiýalara bömekden alynýan peýda siziň olary aýratynlykda ýazyp we sazlap bilýändigizden ybaratdyr. Goşmaça artykmaçlyklar hem bardyr, olar belli bir wezipeleri ýerine ýetirmek üçin ýazylan funksiýalaryň beýleki programmalarda gaýtadan peýdalanyň biljekdiginden ybaratdyr. C++ bilen toplumda üpjün edilýan kitaphanalar programmalaryňyzda ulanyň boljak standart funksiýalaryň köp sanlysyny teklipeň edýarlär. Olar köp möçberdäki gaýtalanyp duran işlerin agramly böleginden halas edip bilerler. Ýonekeý mysal! programmanyň elementlerine gowy düşünmäge kömek eder. Täze taslamany döretmekden başlan bu amaly çaltlandyrmak üçin **Ctrl+Shift+N** klawislerini utgaşygyndan peýdalanyň bilersiniz. 4.2 suratda görkezilen New Project (Täze taslama) dialog penjiresi peýda bolan mahalynda taslamanyň tipi hökmünde Win32 goşundyny we şablon hökmünde Win32 Console Application (Win 32 konsol goşundysy) goşundyny saýlap alyň. Taslamany `Ex2_01` diýip atlandyryň.

Eger indi dialog penjiresiniň çep bölegindäki Application Settings (Goşundylary sazlamak) salgylanma basylanda Win32 goşundysynyň goşmaça parametleri görüp bolar (4.4 surat).

Duýdurmazdan Console application (Konsol goşundysynyň) geçirijisi goýlar, bu bolsa `main()` funksiýasynyň wersiýasyny öz icine alýan faýly göz önünde tutýar, ýöne siz taslamanyň iň ýonekeý guşuşyndan başlamalysyňyz, sonun üçin hem goşmaça parametrleriň toplumynda Empty project (Boş taslama) baýdagyny goýuň we Finish (Taýýar) duwmä basyň. Siziň taslamaňyz döredildi, ýöne onda hiç hili faýl ýok.

Taslamanyň icine nâme giryändigini, 4.5 suratda görkezilişi ýaly, Visual C++2010 işlap düzüş gurşawynyň baş penjiresiniň çep bölegindäki çözgütler geçirilişiniň panelinde görüp bolar.

Siziň boş taslamadan baslaýandygyňyz sebäpli, onuň hasiýetleriniň anyklanyşynyň (bellenişinin) dogrudygyna göz ýetirmek gerek. Menýudan Project-Properties (Taslama-Häsiýetler) nokadyny saýlap ýa-da **Alt+F7** klawisleri utgaşdyryp basmak bilen, taslamanyň hasiýetleri dialog penjiresini açyn. Çep

panelde Configuration (Konfigurasiya) nokadyny, sonra bolsa, eger ol saýlanyp alynmadyk bolsa, General (Umumy) nokadyny saýlap alyň. Sagdaky panelde hasiýetlerin sanawy peýda bolar. Project Defaults (Taslamanyň duýdurmazdan ahmiýeti) kategoriýasyndan Charaçter Set (Simwollar toplumy) nokady saýlaň. Hasiýetlerin adynyň sag tarapyndan açylyan sanaw bolýar, ondan Not Set (Oturdylmady) nokadyny saýlap almaly. Bu hasiýetin Win32 konsol goşundysy üçin duýdurmazdan ahmiýeti siziň goşundyda Unicode simwollar toplumyndan peýdalanmakçy bolýandygynyzy göz önünde tutýar. Netijede kompilýator _vmain funksiýasyny gözlär, bu bolsa ýokdur. Eger siz taslamanyň bu hasiýetini uýtgetmeseňiz, onda kompilýasiya wagtynda ýalňyslyk barada habar alarsyňyz, sebäbi wmain funksiýasy tapylmaz. Charaçter Set hasiýeti goýlandan soň programmanyň koduny döretmek üçin ähli zat taýýar bolýar.

Çözümleri geçirijiniň panelinde Source Files (Başlangyç faýllar) bukjasynda syçanjygyn sag duwmesine basyp täze faýlyň taslamasynyň başlangyç koduny goşmakdan başlan we menýuda Add-New Item... (Goşmaly-Taze element) nokadyny saylan. 4.6 suratda görkezilene meňzes, Add-New Item (Goşmaly-Taze element) dialog penjiresi açylar. C++File (.cpp) (Faýl C++ (.cpp) şablonyn çykandygyna göz ýetiriň, onuň üstüne basyp, 4.6 suratda görkezilişi ýaly, faýlyň adyny giriziň. Faýl awtomatiki usulda .cpp giňelmesini alar, sonuň üçin hem ony ýazmaly bolmaz. Ýalyň adynyň taslamanyň ady bilen gabat gelen halatynda hem hiç hili problēma bolmaz. Taslamanyň faýlynyň .vcproj giňelmesi bardyr, şol sebäbe göre hem onuň ady doly başlangyç faýlyň adyndan tapawutlanar.

Faýl döretmek üçin Add (Goşmaly) duwmesine basyň. Soňra IDE redaktor penjiresiniň panelinde şu kódy giriziň.

```
// Ex2 0] .cpp:
// Programmanyň ýonekeý mysaly
#include <iostream>

using std: :
cout; using std: :
: cout;

int main ( )
{
    int apples oranges; //iki bitin
    sanly uýtgeýan ululygy yglan etmek
    int
    fruit; //...
    ýene biri
    apples = 5; oranges =
    6; //Başlangyç ähmiýetleri
    bermek
    fruit = apples +
    oranges; //Jemini almak
    cout << endl;
    cout << „Apelsin ýeke-tak miwe däldir..." << endl;
```

```

        << „- bizae jemi" << fruit << „mrwe bar";
cout << endl; // Taze setiriň simwolyňy çykarmak

        return 0; //
Programmadan çykamak
}

```

Ex2_01 .cpp kodunyň fragmenti.

Bu mysalyň maksady C++ operatorlaryny ýazmak üçin ulanylýan käbir usullary görkezmekden ybaratdyr, ýöne bu usullar programmirlmegiň gowy stili bolup durmaýar.

Giňelmegiň netijesinde faýlyň C++ başlangyç kody hökmünde identifikasiya edilen badyna, redaktor tarapyndan tanalýan esasy sözler degişli reňke boýalýar. Siz int girizmeli ýerinde Int giritseňiz ýalňyslygy dessine görersizin, sebäbi Int sözi başlangyç kodda dilin esasy sözlerini bölüp görkezmek üçin ulanylýan reňke boýalmaýar.

Eger siz öz taslamaňyzyň açylan ýeri bolan çözgütleri geçiriji paneline seretseňiz (ony görkezmek üçin Ctrl+Alt+L duwmesine basmaly), onda başlangyç kodun täze döredilen faýlyny görersiniz. Çözgütleri geçiriji hemişe taslamanyň faýllaryny görkezýär.

Eger Class View (KJaslary görkezmek) goşundysyna ýa-da Ctrl+Shift+Ç duwmesine bassaňyz Class View görner. Bu panel iki bölekden durýar: global funksiýalary we taslamada kesgitlenen makroslary (şeyle hem, eger taslama öz icine klaslary alýan bolsa, klaslary) görkezýän ýokarky bölekden we aşaky (başdan boş duran) bölekden durýar.

Eger siz 4.7 suratda görkezilişi ýaly, klaslaryň görkezilyän panelinde Global Functions and Variables (Global funksiýalar we üytgap durýan görkezijiler) elementini saýlap alsaňyz main() funksiýasy aşaky panelde peyda bolar. Munuň nämäni aňladýandygy men biraz soň düşündirerin, hazirlikçe bolsa programmanyň islendik ýerinde elýeter bolan funksiýalaryň we/ýa-da uytgap durýan görkezijileriň ählisiniň hakykatdan global bolup durýandyklaryny belläp geçeyin.

Klaslaryň görkezmekde üçünji goýlan sahypa Property Manaqer (Hasiýetleriň dispetçeri) hökmünde bellenendir we eger siz onuň üstüne bassaňyz we + simwollara basmak bilen daragtyň şahalaryny ýaýbaňlandyrsaňyz, onda ol 4.8 suratda görkezilişi ýaly görner.

Bu ýerde döredip boljak iki wersiýa görkezilyär - barlamak üçin sazlaýyş wersiýasy (Debug) we ahyrky sarp edijä ibermek üçin iş wersiýasy (Release). Taslamanyň her bir wersiýanyň hasiýeti görkezilendir we olaryň her birine iki gezek basylandan soň Property Pages (Hasiýetler sahypalary) dialog penjiresi açylar, olarda zerurlyga göre hasiýetleri üytgediler.

Programmany kompilyasiya etmegiň we düzmegiň üç usuly bardyr: Build (Yygnamak) menýusynda Build Ex2_01 (Yygnamak Ex2_01) nokadyny saýlap, F7 klawisine ýa-da gurallar panelindäki degişli duwmä (ony duwmaniň üstünde kursory biraz haýal aýlap tapyp bolar) basmaly. Eger gurallar paneliniň cursor üstüne eltilende yüze çykyan Build Ex2_01 görkezýän duwmesi ýok bolsa, diýmek,

Buüd gurallar paneliniň häzirki pursatda görünmeyandigi bolup durýar. Bu örän uzyn sanaw bolup, siz nâme bilen meşgullanýandygyňyza baglylykda, ondan gurallar paneliniň beýleki toplumlaryny saýlap almagy karar edip bilersiniz.

Eger programmany ygnamak amaly üstünlikli tamamlanypdyr diýsek, onda Strl+F5 klawislerini utgaşdyryp basyp ýa-da Debug (Sazlamak) menýusyndan Start Without Debugging (Sazlaman işe goýbermçk) nokadyny saýlap almak bilen ony işledip bolar. Netijede siz buýruk setiriniň penjiresinde şu netijäni alarsyňyz.

```
Apelsinler ýeke-täk miwe dälidir ....
```

```
Bizde jemi 11 miwe bar.
```

```
Dowam etmek üçin islendik klawisi basyň...
```

Birinji iki setiri programma berdi, soňkusy bolsa siziň ýerine ýetirmegi tamamlap we buýruklar setiriniň penjiresini ýapyp biljekdigiňizi görkezýär. Biz beýleki konsol programmalarynda netijäniň soňky setirini görkezip durmaly, yöne onuň hemişe bardygyny unutmaň.

4.2. Programmada teswirler

Programmanyň birinji iki setiri - bu *teswhierdir*. Teswirler programmanyň möhüm bölegi hasap edilýar, yöne olar ýerine ýetirilýan kod bolup durmaýar - bu dine okyjylar/adamlar üçin ýatlatmalydyr. Teswirleriň ählisi kompilyator tarapyndan inkâr edilýar. Koduň islendik setirinde yzly-yzyna gelýan iki ýapgyt çyzyk (/ /), eger olar tekst setirinde (tekst setirleriniň nämedigini siz soň bilersiniz) ýok bolsa, setirlerin şu çyzyklaryn sag tarapyndaky bölekleriniň teswirler bolup durýandygyny habar berýar.

Görşünüz ýaly, programmadaky birnäçe setirler programmanyň özüniň operatoiriary bilen teswirleri öz icine alýar. Siz hem olary /* we */ simwollar bilen çaklendirip, teswirleriň alternatiw görnüşini ulanyp bilersiniz. Mysal üçin, programmanyň birinji setiri şunun ýaly bolup biler.

```
/* Ex2_01.cpp */
```

// simwollaryndan baslanýan teswir dine bir setiriň onuň sag tarapyndaky böleklerini öz icine almak, şol bir wagtyň özünde /*...*/ görnüşinde /* we */ arasyndaky ähli zatlar teswir bolup durýar diýip kesgitleýar we birnäçe setire degişti bolup biler. Mysal üçin, şunun ýaly ýazyp bolar. /*

```
Ex2_01.cpp
```

```
Programmanyň ýonekeý
```

```
mysaly */
```

Bu ýerde dört setiriň äh 1 i si teswir bolup durýar. setirlerini bölüp görkezmek isleseňiz, onda onylary hesişe bolar:

Eger teswirleriň kabir çarçuwanyň icine alyp

```
*Ex2_01.cpp
```

```
*
```

```
Programmanyň ýonekeý mysaly *
```

Düzgün bolşy ýaly, hemişe programmalary doly teswirlerler bilen üpjün etmäge çalyşmaly. Teswirler beýleki programmaçynyň ýa-da özünizin birnäçe wagtdan soň koduň anyk bir böleginiň maksadyna we onuň nähili isleyändigine düşünp bilmeginiz üçin ýeterlik bolmalydyr. Men has jikme-jik düşündirmek üçin mysal larda teswirleri hakyky programmai ardaka garanynda has ýygy peýdalanaryn.

`ftinclude direktiwasy - sözbaşy faýllary`

Programmada başlangyç teswirleriň yzýanyndan `//include direktiwasy` ýerleşýär.

`#include <iostream>`

Ol kompilyatora nâme etmelidigini - häzirki ýagdaýda bolsa burç şekilli ýaý icinde (<>) görkezilen `iostream` faýlynyň icindakileri ony kompilyasiýa etmezden ön programmanyň başlangyç tekstine goýmalydygyny görkezyändigini sebäpli *direktiwa* diýlip atlandyrylýar. `iostream` faýly sözbaşy faýly diýip atlandyrylýar, sebäbi ol adatça programma faýlynyň başynda peýda bolýar. `iostream` sözbaşy faýly C++ dilinde giriş we çykyş operatorlaryny peýdalanyp bilinmegi üçin gerek bolup biljek kesgitlemeleri öz icine alýar. Eger programma `iostream` faýlynyň icindakileri goşmasan, onda kompilyasiýa edilende ýalňyslyk bolar, sebäbi onda şu faýldaky käbir kesgitlemelere bagly bolan operatorlar peýdalanylýar. Visual C++2010 işlap düzüş gurşawvyndan berilyän köp sanly sözbaşy faýllary bardyr, olar mümkinçiliklerin giň toparyny öz icine alýar. Diliň serişdelerini öwrenmegin barşynda siz olaryň köpüsi bilen tanyşyp bilersiniz.

`//include direktiwasy` bilen goýlup bilinjek sözbaşy faýlynyň adyny burç şekilli ýaý icinde däl, eýsem aşakdaky ýaly görnüşde goşa dyrnak icinde hem görkezyaier.

`//include <iostream>`

Faýlyň adyny goşa dyrnak icinde görkezen wagtyňyzda kompilyator ony ilkibaşda *direktiwa* bolan başlangyç koduň faýlyny öz icine alýan katalogda gözleyär. Eger ol ýerde sözbaşy faýly ýok bolsa, kompilyator ony sözbaşylaryn standart faýllary saklanýan kataloglardan gözleyär.

Eger faýlyň ady burç şekilli ýaý icinde görkezilen bolsa, kompilyator ony dine sözbaşylaryn standart faýllarynda gözleyär. Şeýlelikde, eger başlangyç koda standart sözbaşyny goşjak bolsaňyz, onda ony burç şekilli ýaý icinde ýerlesdiriň, çünki şeýle bolanda ol tiz tapylar. Eger beýleki sözbaşy faýllaryny, düzgün bolşy ýaly, özbaşdak döredilen faýllary goşyan bolsaňyz, olaryň adyny goşa dyrnak icinde ýerlesdiriň; şeýle bolmasa olary asla tapyp bolmaz.

ttinclude operatory - bu preprocessorýň direktiwalarynyň biridir. Preprocessorýň beýleki direktiwalary şu kitapda olaryň yüze çykyşyna göre görkeziler. Visual C++ redaktory olary tanaýar we redaktirlenýan penjirede olary gök reňke boýaýar. *Preprocessorýň direktiwalary* - bu kompilyatoryň desgapky islöp geçýän döwründe ýerine ýetirilýan buýruklardyr, deslapky işlap geçmek başlangyç kod obýekt koduna kompilyasiýa edilmänkä ýerine ýetirilýar,

preprocessor'nyň direktiwalary bolsa başlangyç kod kompilyasiýa edilmezinden ön oňa belli bir derejede täsir edýär.

Atlar ginişligi we usingi yglan etmek

1-nji bapda görkezilişi ýaly, *standart kitaphana* - bu ýygy gabat gelýan wezipeleriň köpüsini çözmek üçin niýetlenen proseduralaryň giň toplumy bolup durýar: mysal üçin, giriş-çykyşy işlap geçmek we binýatlyk matematiği hasaplamalary ýerine ýetirmek. Sular ýaly proseduralaryň örän köp möçberiniň bardygy sebäpli, solar bilen bir hatarda beýleki atlandyrylan barlyklaryň bardygy sebäpli, sizin öz maksatlaryňyz üçin tötänden standart kitaphanada kesgitlenen atlaryň biri bilen gabat gelýan ady peýdalanmagyňyz mümkindir. Atlar ginişligi (namespace) - bu C++ diliniň programmada dürli maksatlar üçin ulanylýan atlaryň gaýtalanmagy bilen bagly problemalaryň önüni almak üçin niýetlenen mehanizmidir. Munuň üçin standart kitaphanadaky atlar ýaly atlaryň belli bir köplügi maşgalanyň umumy ady bilen assosirlenýär, şol hem atlar ginişligi bolup durýar.

Atlar ginişliginde peýda bolýan kodda bellenen her bir at şol giňisligiň adyny öz icine alýar. ISO/IEC C++ standart kitaphananyň ähli serişdeleri std atlar ginişliginde kesgitlenendir, sonun üçin hem standart kitaphananyň siziň öz programmaňyzda yüzlenip biljek. her bir elementiniziň hususy ady we klafikator hökmünde atlar giňisliginiň ady (std) bolýar. cout we endl atlary standart kitaphanada kesgitlenendir, sonun üçin hem olaryň doly ady std: :cout we std: :endl görnüşde bolýar. Giňisligiň adyny elementin adyndan bölüp aýyrýan iki goşa nokat operatory emele getirýär, ol görünyan ýaýlany aýdyňlasdyryjy operator diýlip atlandyrylýar. Soňra men onun peýdalanylýan beýleki ýagdaýlary barada hem gürrün bererin. Programmada doly ady peýdalanmak kódy biraz agyrlasdyrýar, şol sebäpli klafikatory - atlar giňisliginiň adyny (std) görkezmezden ýonekeý atlar peýdalanylsa gowy bolar. Programmanyň //include <iostream> direktiwasyňyň yzyndan gelýan iki setir agzalan mümkinçiligi üpjün edýär.

```
using std: :cout
```

```
using std: :endl
```

Bu - usingi yglan etmedir, ol sizin atlar giňisliginiň adyny görkezmezden std giňisligindaki cout we endl atlaryny peýdalanmak maksadyňyz barada kompilyatora habar berýar. Sondaň soň kompilyator, programmanyň tekstinde cout adyna gabat gelende birinji bildirişe (yglan etme) laýyklykda standart kitaphanada kesgitlenen cout adyny göz önünde tutýandygyňyzy çen edýar. cout akymyň standart çykyşy bolmak bilen, ol duýdurmazdan buýruk setirine laýyk gelýar, endl ady bolsa täze setirin simwolydyr.

Biraz sonra siz bu barada hem-de öz atlar giňisliginiň nähili kesgitlemedigi barada köp zat bilersiniz.

```
main() funksiýasy
```

Soňky mysaida main() funksiýasy main() sözbaşydan we açýan şekilli ýaýdan ({}) başlap degişli ýapýan şekilli ýaýa (}) çenli aralykdaky ähli zatlardan durýar. Şekilli ýaýlar funksiýanyň ýerine ýetirilýan operatorlaryny öz icine alýar, olaryň bary bilelikde funksiýanyň bedeni diýip atlandyrylýar.

islendik funksiýa (beýlekiler bilen bir hatarda) onuň adyny kesgitleýän sözbaşydan durýar, onuň yzyndan funksiýanyň bedeni gelýar, bu jübüt ýaýlaryň icindaki köpsanly operatorlary öz icine alýar. Funksiýanyň esasy bölegi hiç hili operatorlary icine alman biler, ýöne şeýle ýagdaýda funksiýa hiç zat etmeyär.

Hiç zat etmeýän funksiýa artykmaç ýaly bolup görner, ýöne sizin uly programna ýazýan wagtyňyz ilki başda programmanyň doly gurluşyny funksiýada görkezip bilersiniz, ýöne olaryň bedenini boş goýup ýa-da oňa az möçberdäki funksiýa goýup hem bolar. Şeýle etmek bilen siz kompilýasiýa etmek we sonundan işlap taýýarlamagyň barşynda bu funksiýalaryň kodlaryny inkrement goşmak bilen, islendik wagtda tutuş programmany onuň ähli funksiýalary bilen ýerine ýetirmek mümkinçiligini üpjün edýarsınız.

4.3. Programmanyň operatorlary we tesimler

Programmanyň `main()` funksiýanyň bedenini emele getiryän operatorlarynyň her biri *nokatly otur* bilen jemlenýär. Bu simwol setiriň däl-de, operatoryň soňuny belleýar. Diýmek, bir operator, eger bu koda düşünmage kömek edýän bolsa, birnäçe setire degişli bolup biler ýa-da birnäçe operatorlar bir setirde jemlenip bilerler. Programmanyň operatory programmanyň nâme edýändigini kesgitleýän binýatlyk elementdir. Ol belli bir derejede tekstiň tesimindäki sözleme menzeşdir, şunda tesimdäki sözlemlerin her biri hereketi yada ideýany aňladýar, ýöne şunda has umumy ideýany kesgitlemek üçin tesimiň beýleki sözlemleri bilen utgaşýar. *Operator* - kompýuteriň ýerine ýetirip biljek, ýöne has çylşyrymly hereketleri ýa-da hasaplamalary kesgitlemek üçin beýleki operatorlar bilen utgaşyp biljek hereketleriň özüne ýeterlik kesgitlemesidir.

Funksiýanyň hereketleri hemişe operatorlaryň toplumy bilen aňladylýar, olaryň her biri nokatly otur bilen tamamlanýar. Olaryň nähili isleyändiklerini duýmak üçin soňky mysaldaky operatorlara seredip görün. Bu operatorlaryň her biri aşakda has giniş beýan ediler.

```
main() funksiýanyň bedenindeki birinji operator.  
int ales, oranges; // iki bitin sanly uýtgeýan  
ululyklary yglan etmek
```

Bu operator iki uýtgeýan ululygy - `apples`, `oranges` - yglan edýar. Uýtgeýan ululyklar - bu kompýuteriň ýadyndaky at berlen fragment bolup, ony maglumatlary saklamak üçin peýdalanyp bolýar, bir ýa-da ondan köp uýtgeýan ululyklaryň atlaryndan durýan operator bolsa, yglan edilen uýtgeýan ululyklar diýlip atlandyrylýar. Esasy söz `int` mundan önki mysalda `apples`, `oranges` uýtgeýan ululyklaryň bitin san ahmiýetlerini saklamak üçin niýetlenendigini aňladýar. Programmada täze uýtgeýan ululyklaryň adynyň yglan edýan ähli ýerinde hemişe onuň saklap biljek maglumatlarynyň görnüşi görkezilyär, bu hem uýtgeýan ululyklaryň tipi diýlip atlandyrylýar.

Indiki operator beýleki bitin sandaky uýtgeýan ululyklary yglan edýar.
`int fruit; // ... ýene biri`

Bir operatorda birnäçe uýtgeýan ululyklary yglan edip bolýan hem bolsa (mundan önki setirde apples, oranges uýtgeýan ululyklar bilen edilişi ýaly), onuň maksadyny beýan etmek bilen, ony özbaşdak teswirläp bolar ýaly, her bir uýtgeýan ululyk aýratyn operatorda we aýratyn setirde yglan edilse gowy bolar.

Mysalda indiki setir şunun ýaly bolar:

```
apples = 5; oranges = 6; // Başlangyç ahmiýetleri  
bermek
```

Bu setirde iki operator bolýar, şunda olaryň her biri nokatly otur bilen jemlenýar. Men olary bu ýerde dine siziň bir setirde birnäçe operatorlary ýerleşdirip biljekdigiňizi görkezmek üçin ýerleşdirdim. Bu hökman däl hem bolsa, setirde bir operatoryň bolmagy programmirlemekde gowy ähen hasap edilyär, sebäbi bu koda düşünmegi ýeňillesdirýar. Programmirlemekde gowy ähen koduň oňa düşünmegi ýeňillesdirýan we ýalňyslaryň ähtimallygyny peseldýän stihni göz önünde tutýar.

Önki setirdäki iki operator apples, oranges uýtgeýan ululyklaryň, degişlilikde 5 we 6 ähmiýetini saklap galýar, Bu operatorlar *ähmiýet beryän operatorlar* diýlip atlandyrylýar, sebäbi olar uýtgeýanleriň täze ahmiýetlerini beryärler, + - bellik bolsa ähmiýet beryän operatordyr.

Soňa şunun ýaly operator gidyär.

```
fruit = apples + oranges; // Jemini almak
```

Bu hem ähmiýet beryän operatordyr, yöne biraz tapawutlanýar, sebäbi ähmiýet beryän operatoryň sag tarapynda arifmetiki anlatma bar. Bu operator apples, oranges uýtgeýan ululyklarda saklanýan ahmiýetleri jemleýar, soňra bolsa, fruit uýtgeýan ululykda netijäni saklaýar. Sondan soň ýene iki operator gelyär

```
cout << endl; // Çykyşy täze setirden  
başlamak  
cout << „Apelsin ýeke-tak miwe dälendir . . . , „  
endl << „- bizde jemi" << fruit << „ miwe  
bar.";  
cout << endl; // Täze setiriň simwolyny  
çykarmak
```

Bu çykyş operatorydyr. Olaryň birinjisi täze setiriň simwolyny (endl) krandaky buýruk setirine iberýar. C++ dilinde girişiň çeşmesi we çykyşyň bellenen maksady akym (stream) diýlip atlandyrylýar. cout ady „standart" çykyş akymyny görkezýär, « operatory bolsa özüniň sag tarapynda ýerleşýan, cout çykyş akymyna gönükdirilmeli zatlaryň ählisini görkezýär. « operatory maglumat akymalarynyň ugruny görkezýär - lýtgeýan ululykdan ýa-da sagdaky akymdan çep tarapda ýerleşýan bellenen çykyş nokadyna ugry görkezýär. Seýlelikde, birinji operatorda endl ady bilen berlen, täze setiriň simwolyny aňladýan cout sözi bilen

indentifikasiya edilen akyma iberilýar. Netijede cout akymyna geçirilen maglumatlar buýruk setirinde çykarylýar.

cout adynyň we «operatorynyň ahmiýeti siziň ilki başda #include direktiwasyň kömegi bilen programmanyň koduna goşan standart kitaphananyň iostream sözbaşy faýlyňyzda kesgitlenýar. Bu ýerde cout - standart kitaphanadan alnan atdyr, sebäbi ol std atlar ginişligine degişlidir. Eýýam belleniş geçilişi ýaly, #include direktiwasy bolmasa, eger ony - std: :cout görnüşinde doly görkezilmese kompilyator ady tanap bilmeýar. cout adynyň akymyň standart çykyşyny görkezmek üçin niýetlenendigi sebäpli, siz ony başga maksatlar üçin peýdalanmaly dalsiňiz, sona görä-de ony, mysal üçin, üýtgap durýan görkezijiniň ady hökmünde ulanyp bilmersiňiz. Elbetde, düii maksatlar üçin şol bir ady peýdalanmagyň bulaşyklyga sebäp boljakdygy görnüp dur.

Üç çykyş operatoryň ikinjisi iki setire degişlidir:

```
cout << "Apelsin ýeke-tak miwe дә.1 ..." << endl  
<< „- bizde jemi“<<fruit<< "miwe bar.";
```

Ozal belleniş geçilişi ýaly, eger bu kódy has aýdyňlasdyrýan bolsa, siz programmanyň bir operatoryny näçe setire ýerlesdiresiňiz gelse şonça setire ýerleşdirip bilersiniz. Operatoryň soňy hemişe setiriň soňy bilen дә-de, nokatly otur bilen bellenyär. Yzygiderli setirler kompilyator tarapyndan bir operatora okalýar we ol operatoryň adyny aňladýan nokatly otury tapýança kombininirlenýar. Elbetde, bu operatoryň soňunda nokatly otury goýmagy ýatdan çykaran ýagdaýyňyzda kompilyator indiki setiri şol operatoryň bölegi hasap eder we olary birleşdirer. Adatça sonuň netijesinde kompilyatoryň düşüniş bilmeýan ýagdaýy yuze çykýar, şol sebäpli hem siz ýalňyslyk barada habar alarsyňyz.

Yokarda getirilen operator buýruk setirine „Apelsin ýeke-tak miwe дәl ..." diýen teksti iberýar, onuň yzyndan täze setiriň ýene bir simwoly (endl), soňra tekstiň ýene bir bölegi - „- bizde jemi „ - bölegi iberilýar, ondan soň fruit üýtgap durýan görkezijiniň ahmiýeti we jemleýji tekst gelýar. Çykyş akymyna iberilýan elementlerin şunun ýaly yzygiderlilikinde hiç hili problēma ýokdur. Operator çep tarapda ýerine ýetirilýar, şunda her bir element nobat boyunça çykyşa iberilýar. Şunda çykyşa iberilen her bir elementin önünde özüniň « operatorynyň gelyändigine üns beriň.

Üçünji we iň soňky çykyş operatory täze setiriň ýene bir simwolyny ekrana iberýar. Seýlelikde, üç operator programmanyň çykyşyny emele getirýarlar, siz ony görüp bilýarsiňiz.

Iň soňky operator şunun ýaly göriniýar.

```
return 0; // Programmadan çykamak
```

Bu operator main() funksiýasynyň ýerine ýetirilmegini bes edýar we programmany tamamlýar. Dolandyryş amal ulgarrtyna gaýdyp gelýar we gaýdys kódy 0 amal ulgamyna goşundynyn öz wezipesini çözendən son üstünlikli tamamlanandygyny habar beryär. Bu operator barada soň gürrün berler.

Programmanyň operatorlary, eger yörite dolandyryş operatorlary olary ýerine ýetimegiň tertibini uýtgetmedik bolsalar, olaryň ýazylan tertibinde ýerine

ýetirilýar. 3-nji bapda siz ýerine ýetirmegin yzygiderliligini uýtgedýan operatorlar bilen tanyşdynyň.

Tesimler

Tesimler (whitespace) - C++ dilinde probelleriň, tabulýasiýalaryň, täze setiriň, täze sahypanyň we teswirleriň simwollaryny anlatmak üçin peýdalanylýan adalga. Tesimler operatoryň bir bölegi bilen beýleki böleginiň arasyňy bölüji bolup hyzmat edýar we kompilýatora operatoryň int ýaly elementiniň niredede tamamlandygyny we beýlekisiniň nireden baslanýandygyny identifikasiýa etmäge mümkinçilik beryär. Galan ýagdaýlarda tesimler inkär edilýar we hiç hili täsir etmeyär.

Mysal üçin, şu operatora seredip geçelin.

```
int fruit; // ... ýene biri
```

int we fruit elementleriniň arasynda kompilýatoryň olary tapawutlandyryp bilmege üçin iň bolmanda bir tesim (adatça probel) bolmalydyr, yöne probeller kân bolsa, olar inkär edilýar.

Beýleki tarapdan şunun ýaly operatora seredeliň.

```
fruit = apples + oranges; // jemini çykarmak
```

Bu ýerde *fruit* we *=* elementleriniň arasynda, şeýle hem *=* we *apples* elementleriniň arasynda hiç hili tesimleriň bolmagy hökman däl, yöne olary islege göre bu ýere goşubam bolar. Sebäbi *=* operatory - bu harp hem, san hem däl, sonun üçin hem kompilýator ony daş-töwerekden tapawutlandyryp biler. Edil sonun ýaly, *+* alamatynyň töwereginde hem hiç hili tesim gerek däl, yöne siz kódy has okamaga ýaramly edesiniz gelse olary goşup bilersiniz.

Aýdylyp geçilişi ýaly, operatoryň başgaça bolanda bulaşyp biljek elementleriniň arasynda bölüji hökmünde ulanylmagyndan başga ýagdaýda tesimler kompilýator tarapyndan inkär edilýar (elbetde, eger olar iki sany goşa dyrnagynyň arasyndaky sertirleriň bir bölegi bolup durmayan bolsa). Sonun üçin hem öz programmaňyza işleginize göre, eger bu ony okamaga has ýaramly edýar diýip hasap edýan bolsaňyz, tesimleri goşup bilersiniz, bir operator programmanyň birnäçe setirine degişli edilen ýokarda getirilen mysalda şeýle edilipdi. C++ dilinde operatoryň soňunyň nokatly otur bolup durýandygyny ýatdan çykarmaň.

Operatorlaryň **bloklary**

Siz birnäçe operatorlary şekilli ýaýlaryň icine alyp bilersiniz. Şu ýagdaýda olar blogy ya-da düzüm operatoryny emele getirýarlar. Funksiýanyň bedeni blogyň mysaly bolup biler. Şunun ýaly düzüm operatoryna bir operator hökmünde seredip bolar (siz 3-nji bapda C++ dilindäki operatory owreneňizde muňa göz ýetirensiňiz). C++ dilindäki aýratyn operatoryň ýerlesdirilip bilinjek ähli ýerinde şekilli ýaýlaryň icinde operatorlaryň blogynyň ulanylmagyna ýol berilýar. Sonuň netijesi hökmünde bloklar beýleki bloklara goşulyp bilner. Elakýatdan bloklar biri-biriniň icine islendik çunlukda goýlup biler.

Bellik. Operatorlar blogy uýtgap durýan görkezijlere hem möhüm täsir edýar, yöne men bu tema degişli gürrüni uýtgap durýan görkezijlerin gömüşi baradaky temanyň gozgalýan pursatyna çenli soňa goýýaryn.

4.4. Awtomatiki usulda döredilýän konsol programmalary

Soňky mysalda biz başlangyç faýllarsyz boş taslama döretdik, sonra kem-kemden faýllary goşduk. Eger siz 1-nji bapda edilişi ýaly, Application Wizard goşundylar ussasyna taslama döretmäge mümkinçilik berseňiz, onda taslama ilki başdan bimaçe faýllary öz icine alar we size ondaky ähli zatlary biraz çun öwrenmek gerek bolar. Ex2_01A atly Win32 konsol programmasynyň täze taslamasyny döredin, ýöne bu gezek Application Settings (Goşundylary sazlamak) dialog penjiresinde hiç hili parametieri bellemän Application Wizard goşundylar ussasyna öz işini etmäge mümkinçilik berin. Taslama dört faýly öz icine alar: Ex2_01A .cpp we stdafx .cpp, olar başlangyç kody, sözbaşy faýlyny, stdafx.h, şeýle hem Windows amal ulgamynyň in irki, siziň gosundyňyzy ýerine ýetirmäge ukyply wersiýasyny kesgitleýän targetver faýlyny öz icine alýar. Olar hiç zat etmeýän işleyen konsol programmasyna zerur bolan binýatlyk serişdeleri üpjün etmek üçin niýetlenendir. Eger sizde açyk taslama bar bolsa, File-Close Solution (Faýl-Çözüdi ýapmak) menýusyndaky nokady saýlap, ony ýapyň. Ýa-da önküni yapman täze taslamany döredin - şunda ol, eger siz bar bolan çözüde tazesini goşmasanyz, ol awtomatiki usulda ýapylar.

Ilki başda Ex2_01A .cpp faýlynyň icindakiler sular ýaly bolar.

```
// Ex2_01A .cpp : konsol goşundysynyn giriş nokadyny kesgitleär //
#include „stdafx.h“
int tmain (int argç, TCHAR* argv
[]) { return 0;
}
```

Ex2_01A .cpp kodunyň fragmenti

Bu önki mysaldan duýpli tapawutlanýar. stdafx.h sözbaşy faýly üçin #include direktiwasy bar, önki wersiýada bu ýokdy we programmanyň ýerine ýetirilip baslanýan funksiýasy main() däl-de _tmain() diýlip atlandyrylýar.

Application Wizard goşundylar ussasy taslamanyň bölegi hökmünde stdafx.h sözbaşy faýlyny döretdi we onuň koduna seredip görseniz, onda standart kitaphananyň stdio.h we tchar.h sözbaşy faýllary üçin iki #include direktiwasyyny görersiniz. Şunda stdio.h faýly - bu giriş-çykyş üçin köne stilin sözbaşysydyr, ol häzirki ISO/IEC C++ standarty peýda bolmazyndan ön peýdalanylypdy, Ol iostream faýlynyň ýerine ýetirýän funksiýalaryny üpjün edýar. Ikinji tchar.h faýly Microsoftyň käbir tekst funksiýalaryny kesgitleýän yörite sözbaşy faýly bolup durýar. Bu ýerde ideýa stdafx.h faýlynyň siziň taslamaňyz üçin standart ulgamlaryn gosulýan faýllaryň toplumyny kesgitlemelidir, onda #include direktiwasyynyň kömegi bilen özünize gerek bolan beýleki ulgamlaryn faýllaryň islendigini goşup bilersiniz. ISO/IEC C++ dili öwrenilende size stdafx.h faýlyna goşulan sözbaşy faýllarynyň hiç birini peýdalanmak gerek bolmaz, bu bolsa Application Wizard goşundylar ussasynyň duýdurmazdan faýllary awtomatiki usulda döretmek serişdelerini ulanmazlyk üçin bir sebäp bolup durýar.

Ozal düşündirilişi ýaly, Visual C++2010 işlap düzüş gurşawy Unicode simvollaryny peýdalanýan programmalar ýazylanda main() funksiýasyna alternatiwa hökmünde wmain() funksiýasyny goldaýar. main() funksiýasy Microsoft üçin mahsus atdyr, ol ISO/IEC C++ diliniň bir bölegi bolup durmaýar. Muny goldamak üçin tchar.h sözbaşysy _tmain adyny onuň adatça main adyny çalyşmagy ýaly kesgitleýar, ýöne şunda UNICODE simwoly kesgitleneknde wmain çalşylyar. Başga sözler bilen aýdylanda Unicode kodlaşdyrmany peýdalanýan programmany identifikasiýa etmek üçin siz stdafx.h sözbaşy faýlynyň başynda şu operatory goşmaly bolarsyňyz.

```
#ifndef UNICODE
```

Hakykatdan siz ýaňy döredilen Ex2__01 A taslamasynda şeýle etmeli dalsiňiz, sebäbi taslamanyň Character Set (Simwollar toplumy) hasiýeti duýdurmazdan Unicode simwollar toplumyny peýdalanýar.

Şu ýagdaýlar aýdyňlasdyrylandan soň biz konsol goşundylary bolup durýan soňky mysallarda ISO/IEC C++ dili üçin önki main() funksiýasyny peýdalanmaga gaýdyp gelip bileris, sebäbi bu ISO/IEC C++ dilinde bu mümkinçilik standart bolup durýar we şeýlelikde, programmiemegin has geçirip bolýan çemeleşmesinden ybaratdyr.

Uýtgeýan ululyklary kesgitlemek

Ähli kompýuter programmalarynyň esasy bellenen maksady - käbir maglumatlary manipulirlemekden we käbir jogaplary almakdan ybaratdyr. Bu işin möhüm serti siziň ygtyýaryňyzda belli bir derejede oylanyşykly at boýunça ýüzlenip we maglumatlaryň elementini saklap boljak ýadyň bolmagydyr. Ýadyň şu gömüşde kesgitlenen her bir fragmenti uýtgeýan ululyk diýlip atlandyrylýar.

Siziň bilsiniňiz ýaly uýtgeýan ululyklaryň her biri belli bir görnüşdäki maglumatlary saklaýar, saklanýan maglumatlaryň tipi bolsa programmada uýtgeýan ululyk yglan edilende ýazylyp bellenýar. Bir uýtgeýan ululyk dine bitin sanlary saklap bilýar we siz ony droblanan ahmiýetleri saklamak üçin peýdalanyp bilmersiňiz. Eler bir pursatda uýtgeýan ululykda saklanýan anyk ahmiýet siziň programmaňyzyň operatorlary tarapyndan kesgitlenýar we elbetde, düzgün bolşy ýaly, onuň ahmiýeti programmanyň ýerine ýetirýan hasaplamalarynyň barşynda birnäçe gezek üýtgap biler.

Indiki bölümde biz uýtgeýan ululyklar yglan edilende olary atlandyrmagyň kadalary barada gürrün ederis.

Uýtgeýan ululyklary atlandyrmak

Uýtgap durýan görkezijä beryän adyňyz identifikátor ýa-da uýtgeýan ululygyň ady diýlip atlandyrylýar. Uýtgeýan ululyklaryň atlary A-z harplaryny (ýokarky ýa-da aşaky registrlerde), 0-9 sanlary we aýratyn belleme alamatlaryny öz içine alyp biler. Beýleki simvollarýň ulanylmagyna ýol berilmeýar we eger siz tötänden haýsydyr bir başga simwoly görkezen bolsaňyz, onda programmany kompiliiejek bolanyňyzda ýalňyslyk barada habar alarsyňyz. Uýtgeýan ululyklaryň atlary harpdan ýa-da aýratyn bellemek alamatyndan başlanmalydyr. Adatça uýtgeýan ululyklaryň atlaryny saýlap almak olarda saklanýan maglumatlaryň görnüşi bilen şertlendirilendir.

Visual C++2010 islāp düzüş gurşavvynda uýtgeýan ululyklaryň atlarynyň 2048-e çenli simwolynyň bolup biljekdigi sebäpli, uýtgeýan ululyklara at bermekde ýeterlik erkinlik berilýar. Hakykatdan uýtgeýan ululyklardan başga-da, C++ dilinde öz atlary bolan we uýtgeýan ululyklaryň atlarynyň berjaý edýan kadalary ýaly kadalara tabyn bolýan 2048-e çenli simwoly öz icine alyp bilýan birnäçe beýleki zatlar hem bar. Aňrybas uzynlykdaky atlaryň ulanylmagy programmaňyzyň okalmagyny kynlaşdyryp biler we eger siz klawiaturada islemäge örän ökde bolmasaňyz olary ýazmak kyn bolar. Atlar bilen işlenende üns berilmeli ýene bir düýpli ýagdaý - bu ähli kompilýatorlaryň şunun ýaly uzyn atlary goldamaýandygy bolup durýar. Eger öz koduňyzy başga gurşawlarda kompilirlmek maksadyňyz bar bolsa, onda 31 simwoldan geçmeyän at bilen çaklenilse gowy bolar. Adatça bu adyň oylanyşykly ähmiýetini saýlap almak üçin ýeterlikdir we beýleki kompilýatorlara mahsus bolan, atlaryň uzynlygynyň çaklendirilmegi bilen bagly problemalardan gaça durmaga mümkinçilik beryär. Uýtgeýan ululyklaryň aýratyn bellemek alamatyndan (mysal üçin, `_this` we `__that`) baslanýan atlaryny peýdalanmaga rugsat edilyän hem bolsa, ondan gaça durulsa gowy bolar, çünki bu sonuň ýaly görnüşdäki standart uýtgeýan ululyklar bilen konfliktleriň ýüze çykmak ähtimallygy bilen baglydyr. Şol sebäp boýunça uýtgeýan ululygyň iki aýratyn belleme alamatyndan baslanýan atlaryndan hem gaça durmak gerek.

Aşakda uýtgeýan ululyklaryň gowy atlarynyň mysallary getirilýar.

```
pri ke
discout
pShape
value
COUNT
```

8J3all, 7Up we 6__paçk ýaly atlary, edil sonuň ýaly hem Hash! ýa-da Maru-Ann ýaly atlary ulanmaga ýol berilmeýar. Kese çyzygyn ýerine aýratyn belleme alamaty bolan Maru-Ann adynyň kabul ederiikdigine garamazdan, soňky mysal köp ýaýran ýalňyslygy görkezýär. Elbetde, Mary Ann - hem dogry däl, sebäbi uýtgeýan ululyklaryň atlarynda probellere rugsat berilmeýar. Uýtgeýan ululyklaryň republiçan we Republiçan ýaly atlarynyň dürli atlar ýaly düşünilýändigine üns beriň, sebäbi uýtgeýan ululyklaryň atlary registre bagly bolup durýar, ýagny ýokarky we aşaky registriň simwollary biri-birinden tapawutlanýar. Elbetde, tesim simwollary (probel, tabulýasiýa, täze setir) atlarda görkezilip bilinmez we eger siz olary tötänden goşan bolsaňyz, onda bir adyň ýerine iki ýa-da ondan gowrak ady alarsyňyz, bu bolsa adatça kompilýatory degişli habar ibermege mej bur edýär.

C++ dilinde kabul edilen standart ylalaşyk uly harplar bilen başlanan atlaryň klaslar üçin, setir harplaryndan baslanýanlaryň bolsa uýtgeýan ululyklar üçin peýdalanylmagyny göz önünde tutýar. Klaslar barada 8-nji bapda gürrün ediler.

4.5. C++ dilinde esasy sözler

C++ dilinde atyaplykdaky sözler bardyr, olar esasy sözler diýlip atlandyrylýar, dilde olaryň yörite ähmiýeti bardyr. Programmanyň teksti girizilen Visual C++2010 işlap düzüş gurşawynyň redaktory olary belli bir reňk bilen söhlelendiryär, meniň ulgamymda olar duýdurmazdan gök reňke boýalandyr. Eger siziň girizen esasy sözünüz degişli görnüşde boýalmýan bolsa, diýmek, siz ony ýalnys girizensiňiz. Ýeri gelende aýtsak, tekst redaktory tarapyndan peýdalanylýan duýdurmazdan bellenen reňk yaramayan bolsa, Tools (Serwis) menýusynda Options (Parametrler) nokady saýlap almak bilen we Environment /Fonts and Colors (Gurşaw/Sriftler we reňkler) dialog penjiresinde uýtgetmeleri girizmek arkaly olary çalsyn.

C++ dilindäki beýleki zatlar ýaly, esasy sözlerin hem registere baglydygyny ýatda saklaň. Mysal üçin, siziň ozal girizen programmaňyzda `int` we `return` esasy sözler bar diýeliň. Eger siz `Int` ýa-da `Return` görnüşde girizen bolsaňyz, onda bu esasy sözler hasap edilmezdi we kompilyator olary esasy sözler hökmünde tanamazdy. Mundan beyläkki öwrenmegin barşynda C++ diliniň beýleki köp sanly esasy sözleri bilen tanyşyp bilersiniz. Öz programmalaryňyz üçin saýlap alýan atlaryňyz C++ dilindäki esasy sözlerin biri bilen hem gabat gelmeli daldır.

Uýtgeýan ululyklary yglan etmek

Siziň eýýam görüp geçişiniz ýaly, uýtgeýan ululygy yglan etmek - bu uýtgeýan at we tip berýan operatordyr, mysal üçin sular ýaly:

```
int value;
```

Bu bitin sanlary saklamak üçin `value` atly uýtgeýani ululygy yglan edyär. `value` uýtgeýan ululykda saklanyp bilinjek maglumatlaryň tipi `int` esasy söz bilen görkeziler, sonun üçin hem `value` uýtgeýani dine `int` tipli maglumatlary saklamak üçin ulanyp bolar. `int` sözünüň esasy söz bolup durýandygy sebäpli ony biziň uýtgeýan ululyklarymyzyň hiç biriniň ady hökmünde peýdalanyň bolmaýar.

Bellik. Uýtgeýan ululygy yglan etmek hemişe nokatly otur bilen tamam lanýar.

Bir yglan etmede birnäçe uýtgeýan ululygyň adyny görkezip bolar, yöne eýýam aýdylýp geçilişi ýaly, adatça olary aýry-aýry operatorlarda - her setirde birini yglan edilmegi govvy bolar. Şu kitapda men kähalatda bu kadadan çykyaryn, yöne dine kody köp sahypalara süyndirmezlik üçin şeýle edyärin.

Maglumatlary (mysal üçin, bitin sanyň ahmiýetini) saklamak üçin uýtgeýan ululygyň adyny kesgitlemek ýeterlik daldır. Bu at kompýuteriň ýadynyň kabir fragmentlerini baglanyşdyrmak zerurdyr. Bu proses uýtgeýan ulugy kesgitlemek diýip atlandyrylýar, C++ dilinde uýtgeýan ululygy yglan etmek onuň kesgitlemesi hem bolup durýar (käbir yörite ýagdaýlar muňa girmeyär, olar barada soň gürrün ediler). Aýratyn operatorda siz uýtgeýan ululygyň adyny gorkezyarsaňyz, bu bolsa ony ýadyň belli bir möçberdäki uýastogly bilen baglaýar.

Şeýlelikde `int value;` operatory şol bir wagtyň özünde yglan etme we kesgitleme bolup durýar. Bu ýerde kompýuter kompýuteriň ýadynyň uýastogynyň elýeter bolmagy üçin yglan edilen uýtgeýan ululygyň `value` ady peýdalanylypdyr,

ýadyň bu uçastogy şu ýerde kesgitlenýar we int tipli aýratyn ahmiýetini saklap biler.

Bellik. Yglan etme adalgasyny siz bu adyň name üçin peýdalanylýandygy baradaky maglumat bilen bilelikde, programmada at bermek üçin ulanarsyňyz. Kesgitleme adalgasy şol at bilen bagly kompýuter ýadyny bölüp bermegi beýan edýar. Uýtgeýan ululyk bolanda yglan etme bilen kesgitlemäni, ýokarda getirilen koduň setirinde görkezilişi ýaly, bir operatora birleşdirip bilersiniz. Yglan etme bilen kesgitlemäniň arasyňy şunun ýaly takyk tapawutlandyrmagyň sebäbi soň siziň kesgitleme däl-de yglan etme bolup durýan operatorlara gabat geljekdigiňizden ybaratdyr.

Siz uýtgeýan ululygy öz programmaňyzyň başlangyjy bilen onuň birinji gezek ulanylan ýeriniň arasyndaky nokatda yglan edip bilersiniz. C++ dilinde uýtgeýan ululygy onuň birinji gezek ulanylýan ýerine golaý ýerde yglan etmek gowy hasap edilýar.

Uýtgeýan ululygyň **başlangyç** ahmiýetleri

Uýtgeýan ululygy yglan etmek bilen dessine oňa başlangyç ahmiýet berip bilersiniz. Oňa başlangyç ahmiýet berýan uýtgeýan ululygy yglan etmek *in is i al l aş dyrmak* diýlip atlandyrylýar. Yglan edilen wagtynda uýtgeýan ululygy inisiallaşdyrmak üçin deňlik alamatyny girizmek gerek, sonun yzyndan inisiallaşdyrýan ahmiýet gelýar. Her bir uýtgeýan ululyga başlangyç ahmiýet bermek üçin şu operatorlary ýazyp bilersiniz.

```
int value = 0;  
int count = 10;  
int number = 5;
```

Şu ýagdaýda value uýtgeýan ululyk 0 ahmiýeti, count uýtgeýan ululyk 10 ahmiýeti, number uýtgeýan ululyk bolsa 5 ahmiýeti alýar.

Uýtgeýan ululyklaryň başlangyç ahmiýetlerini görkezmegin başga usuly hem bardyr, ol *funksional notasiýa* diýlip atlandyrylýar. Deňlik alamatynyň we ahmiýetiň ýerine siz ýaý icinde uýtgeýan ululygyň sonundan gelýan ahmiýeti gorkezýarsyňiz, ýagny oňki yglan etmeleri şu görnüşde gaýtadan ýazyp bilersiniz.

```
int value = (0);  
int count = (10) ;  
int number = (5) ;
```

Adatça uýtgeýan ululyklar inisiallaşdyrylanda notasiýanyň bir ýa-da beýleki tipini peýdalanyp bolar. Seýle-de bolsa, men siziň iş kodunda olaryň ikisiniň hem nähili ulanylýandygyny görüp bilmeginiz üçin, mysallaryň birinde bir notasiýany, beýleki mysallarda başga notasiýany peýdalanaryn.

Eger siz uýtgeýan ululyga başlangyç ahmiýeti bermeýan bolsaňyz, onda, düzgün bolşy ýaly, ol erkin toplanan „zir-zibili“ öz icine alýar, ol ýadyň uýtgeýan ululyk üçin bölünip berlen uçastogynda bolýar (bu kadadan çykmalar bardyr, şu bapda bu barada durlup geçiler). Sonun üçin hem, onuň mümkin bolan ýerinde uýtgeýan ululyklary olaryň yglan edilen wagtynda inisiallaşdyryň. Eger uýtgeýan ululyklar ilkibaşdan belli ähmiýetleri öz icine alýan bolsa, onda nämädir bir zadyň bolmaýsý ýaly gitmeyändigini anyklamak has ýeňil bolýar. Eger nämädir bir zat

bolmaýsy ýaly gitmeýan bolsa, onda onuň hökmany ýagdaýda bolup geçjekdigine ynanyp bolar.

Maglumatlaryň binýatlyk tipleri

Uýtgeýan ululygyň öz icine alyp biljek maglumatlarynyň köpdürliligi onuň *maglumatlar tipi* diýlip atlandyrylýar. Programmaňyzdaky ähli maglumatlar we uýtgeýan ululyklar maglumatlaryň belli bir tipine degişli bolmalydyr. C++ ISO/IEC standarty belli bir esasy sözler bilen berilýan maglumatlaryň binýatlyk tipleriniň toplumyny teklipl edýar.

Maglumatlaryň binýatlyk tipleri diýlip atlandyrylýandygynyň sebäbi olar siziň kompýuteriňizdäki binýatlyk maglumatlary görkezyän tiplerin ähmiýetlerini - hakykat yüzünde san ähmiýetlerini saklaýar, olara simwollar hem girýär, sebäbi olar san kodlary bilen görkezilýär. Siz eýýam bitin sanly uýtgeýan ululyklary kesgitlemek üçin ulanylýan int esasy sözi gördüňiz. C++/CLI dilinde hem maglumatlaryň binýatlyk tipleri kesgitlenendir, olar ISO/IEC C++ standartynyň bir bölegi bolup durmaýar, olar barada men şu bapda girrün bererin.

Binýatlyk tipler üç kategoriýa bölünýar - bitin sanlary icine alýan tipler, bitin sanly ähmiýetleri icine almayan tipler we ähmiýetleriň boş köplügini ýa-da tipiň ýokdugyny görkezyän void tipi.

Bitin sanly uýtgeýan ululyklar

Aýdylýp geçilişi ýaly, bitin sanly uýtgeýan ululyklar - bu bitin sanlaryň ähmiýetini icine alýan uýtgeýan ululyklardyr. Futbol toparynyň oýunçylarynyň sany - iň bolmanda oýnuň başynda bitin san bolup durýar. Bitin sanly uýtgeýan ululyklary int sözi bilen yglan edip bolýandygyny siz bilýarsňiz. int tipli uýtgeýan ululyklar ýadyň 4 baýtyny tutýar we oňyn hem yaramaz bitin sanly ähmiýetleri saklap bilerler, int tipli uýtgeýan ululyklaryň ýokarky we aşaky çäkleri anrybaş we iň pes goşa ähmiýete laýyk gelip biler, ol 32 baýtdan ybarat bolup biler, int tipli uýtgeýan ululygyň ýokary çägi - $2^{31} - 1$, bu 2 147 483 647 sana deňdir, aşaky çäk bolsa - (2^{31}) deňdir, bu - 2 147 483 648 sana deňdir. Aşakda int tipli uýtgeýan ululygyň mysaly görkezilýär.

```
int toCount = 10;
```

Visual C++2010 işlap düzüş gurşawynda esasy söz short hem bitin sanly uýtgeýan ululyklary kesgitleýar, ýöne olar 2 baýty tutýar. Esasy söz short short int sözüne ekwiwalentdir we siz aşakda görkezilýän operatorlaryň kömegi bilen short tipli iki uýtgeýan ululygy kesgitläp bilersiniz.

```
short feetPerson = 2;
```

```
short int feetPerYard = 3;
```

Uýtgeýan ululyklaryň ikisi hem şol bir tipe degişlidir, sebäbi short sözi short int sözünüň aňladýan zadyny aňladýar. Bu erde men olaryň nähili ulanylýandygyny görkezmek üçin tipiň adynyň iki görnüşini hem peýdalandym, ýöne programma ýazylanda bir wariant bilen çäklenilse gowy bolar, şunda short sözünüň ýygy gabat gelyändigini belläp bolar.

C++ dilinde bitin sanly tip - long hem göz önünde tutulandyr, ony long int ýaly hem ýazyp bolar. long tipli uýtgeýan ululyklar şunun ýaly yglan edilýär.

```
long bigNumber = 1000000L;
```

```
long lagreValue = OL;
```

Elbetde, siz başlangyç ähmiýetleri görkezip funksiýalaryň notasiýasyndan hem peýdalanyň bilersiniz.

```
long bigNumber = (1000000L);
```

```
long lagreValue = (OL);
```

Bu operatorlar başlangyç ähmiýetleri, deňşililikde, 1 000000 we 0 bolan long bigNumber we long lagreValue uýtgeýan ululyklary yglan edýarlar. Literalyň sonuna goşulan E harpy onuň long tipli bitin sandygyny görkezýär. Şu maksatlar bilen 1 setir harpyny hem ulanyň bolar, ýöne ol 1 san ýaly hem okalyp bilner. L harpy goşulmadyk bitin sanly literallaryň int tipi bardyr.

Bellik. Programmada uly sanlar ýazylanda toparlaryň arasyny bölüjiler hökmünde otur ýa-da probel goýulmaly däl. Adaty tekstde 12,345 ýa-da 12 345 ýaly ýazyp bolar, ýöne programmanyň kodunda 12345 diýlip ýazylymalydyr.

long hökmünde yglan edilen uýtgeýan ululyklar Visual C++2010 gurşawynda 4 baýty tutýar we - 2 147 483 648-den - 2 147 483 647-ä çenli ähmiýeti kabul edip biler, int tipli uýtgeýan ululyklar hem şol diapazony kabul edip bilerler.

Bellik. C++ dilindäki beýleki kompilyatorlarda long tipli uýtgeýan ululyklar (long int ýaly) int tipinden tapawutlanyp biler, sonun üçin hem eger siz programmany beýleki gurşawlarda kompilirlmekçi bolsaňyz, onda long we int sözleriniň ekwiwalentdigine bil baglaman. Elakýatdan geçirip bolýan kod ýazylanda siz hatda int tipiniň 4 baýt tutýandygyna hem bil baglamaly dalsiňiz (mysal üçin, Visual C++ gurşawynyň oňki 16 razryadly wersiýalarynda int tipli uýtgeýan ululyklar iki baýtlydy).

Eger ondan hem uly möçberdäki bitin sanlary saklamak gerek bolsa, long long tipli uýtgeýan ululyklardan peýdalanyň.

```
long long huge - 1000000000LL;
```

long long tipli uýtgeýan ululyklar 8 baýty tutýar we 9 223 372 036 854 775 808-den 9 223 372 036 854 775 807-ä çenli ähmiýetleri saklap bilerler, long long tipli bitin sanly konstantany LL ýa-da 11 suffuksi beryär, ýöne olaryň soňkusyndan gaça dursaň gowudyr.

Maglumatlaryň **simwol** tipleri

char maglumatlar tipi iki maksada hyzmat edýar. Ol bir baýtly uýtgeýan ululygy beryär, onda ähmiýetleriň belli bir diapazonynyň çäklerinde bitin sanlary ýa-da ASCII (Ameriçan Standart Çode Information Interçhange - Amerikan maglumatlary alyşmak standart kódy) aýratyn simwolyny saklap bolýar. char tipli uýtgeýan ululygy şu operatoryň kömegi bilen yglan edip bolar.

```
char letter = 'A';
```

Funksiýalaryň notasiýasyny hem peýdalanyň bolar.

char letter = ('A') Bu kod 'A' konstantasy bilen inisiallaşdyrylan letter uýtgeýan ululygy yglan edýar. Ähmiýetiň siziň ozal ekrana çykarynyzda simwol setirleri üçin peýdalanan goşa dyrnak däl-de, bir dyrnak icindäki aýratyn simwol hökmünde görkezilyändigine üns beriň. Simwollar setiri - bu char tipli ähmiýetlileriň yzygiderliligidir, ol *massiw* diýlip atlandyrylýan bir bitewi topara jemlenendir. Massiwlere we C++ dilinde setirlerin işlenip geçilişine 4-nji bapda seredilýär.

ASCII kodlaşdyrmada 'A' simwolyňyň 65 onluk ähmiýeti bilen görkezilendigi sebäpli, siz bu operatory şu görnüşde ýazyp bilersiniz:

```
char letter = 65; // A simwolyňyň ekwiwalenti
```

Bu önki operatoryň beren netijesi ýaly netije berip biler.

Char tipli uýtgeýan ululyklarda saklanyp biljek bitin sanlaryň diapazony Visual C++ gurşawynda -128-den 127-ä çenli bolýar.

Bellik. ISO/IEC C++ standarty char tipiniň bir baýtly bellikli bitin sanlary görkezmegini talap etmeyär. Bu kompilyatoryň durmuşa geçirmek üçin saýlap alşyna baglydyr: char tipini -128-den 127 aralygyndaky diapazonda bellikli bitin sanlar hökmünde görkezmelimi ýa-da 0-dan 255 aralygyndaky diapazonda belliksiz bitin sanlar hökmünde görkezmelimi. Eger özünüziz C++ dilindäki koduňyzy başga gurşawa geçirmekçi bolsaňyz, onda şu ýagdaýy göz önünde tutun.

wchar tipi bu tipiň giň simwollar tipidigi (wide, char type) bu tipiň uýtgeýan ululyklarynyň 0-dan 65 535-e çenli ähmiýeti bolan 2 baýtly simwollary saklaýandygy sebäpli şeýle atlandyrylýar. Aşakda *wchar_t* uýtgeýan ululygy kesgitlemegiň mysaly getirilýar.

```
wchar_t letter = L ' Z '; // Uýtgeýan ululyk simwolyň  
16 bitli koduny saklaýar
```

Bu operator 16 bitli kod tarapyndan Z harpy bilen inisiallaşdyrylan letter uýtgeýan ululygy kesgitleýar. 7⁷ simwol konstantynyň özünden gelýan L simwoly kompilyatora onuň simwolyň 16 bitli simwoly bolup durýandygyny habar berýar. *wchar_t* uýtgeýan ululyk Unicode kodlarynyň ähmiýetini saklaýar.

Bu ýerde funksiýalaryň notasiýasyny peýdalanyp bilersiniz.

```
wchar_t letter = (L + Z4); // Uýtgeýan ululyk simwolyň 16 bitli koduny  
saklaýar
```

char tipli (sonuň ýaly beýleki bitin tiplerin) uýtgeýan ululyklaryny inisiallaşdyrmak üçin on alty sanly konstantany hem peýdalanyp bolar, bu köp ýagdaýda amatly bolýar. On altylyk san onluk ekwiwatlentinde 0-dan 15-e çenli sanlary aňladýan, 0-dan 9-a we A-dan F-e çenli (ýa-da a-dan f-e çenli) on altylyk sanlary standart görkezmegi ulanmak bilen ýazylýar. Mundan başga-da, olary onluk ähmiýetlerden tapawutlandyrmak üçin, olaryň önünden 0x (ýa-da 0X) getirilýar. Seýlelikde, bu netijäni almak üçin, siz ýokarda getirilen operatory şu görnüşde gaýtadan ýazyp bilersiniz.

```
wchar_t letter = (0x5A); // Uýtgeýan ululyk  
simwolyň 16 bitli koduny saklaýar
```

Bellik. Onluk bitin sanly ähmiýetleri önünde nol goýup ýazmaň. Kompilyator olary sekizlik sanlar ýaly interpretirleýar, sonuň üçin hem 065 diýlip ýazylan ähmiýet onluk notasiýada 53-e deňdir. Windows HP, Vista we Windows 7 amal ulgamlarynyň Windowsda elýeter bolan islendik şriltlerde *simwoUary* tapmaga mümkinçilik berýan Charaçter Map (Simwollar tablisasy) utility berýändigine hem üns beriň. Ol on altylyk görnüşde simwollaryň kodyny görkezyär we olary girizmek üçin haýsy klawise basmalydygyny habar berýar. Charaçter Map utilitasyny Start duwmesiniň menýusyndaky System Tools-Aççessorries (Ulgamlaýyn-Standart) papkadan tapyp bolar.

Bitin sanly tiplerin **modifikatorlary**

char, int, short we long bitin sanly tiplerin iiýtgeýan ululyklary bellikli bitin sanly ahmiýetleri (signed) duýdurmazdan saklaýar, sonuň üçin olary oňyn we ýaramaz ahmiýetleri saklamak üçin ulanyp bilersiniz. Sebäbi bu tipler üçin duýdurmazdan signed tipli modifikator kabul edilendir. Sonun üçin int ýa-da long ýazýan wagtyňyzda bu, degişlilikde, signed int ýa-da signed long aňladýar.

Uýtgeýan ululygyň tipini görkezmek üçin signed esasy sözi aýratyn hem peýdalanyp bilersiniz. Şu ýagdaýda ol aşakdaky ýaly, signed int aňladýar.

```
signed value = ~5; // signed int ekwiwalenti
```

Yöne şu hili ýazmak giň ýaýran däl, men hem int peýdalanmagy gowy görýarın.

char tipli uýtgeýan ululykda saklanyp biljek ahmiýetleriň diapazony -128-den +127 aralygynda bolýar, bu bolsa short char tipli uýtgeýan ululyklaryň ýol berilýan ahmiýetlerine laýyk gelýar. Yöne bu nähili gynançly bolsa-da, char we signed char tipleri dürli tipler ýaly düşündirilyar, sonun üçin olar bir zatdyr öydüp ýalňysaýmaň.

Eger siz uýtgeýan ululykda ýaramaz ahmiýetleri saklamagyň gerek bolmajakdygyna ynamyňyz bar bolsa (mysal üçin, oňa bir hepdede geçyan ýoluňyzy (mil) ýazmakçy bolsaňyz), onda uýtgeýan ululygy unsigned ýaly görkezin.

```
unsigned long mileage = OUL;
```

mileage uýtgeýan ululyga ýazyp boljak iň pes ahmiýet nola deňdir, iň ýokary ahmiýet bolsa - 4 294 967 295 (ýagny, $2^{32} - 1$) deňdir. Ony signed long tipi üçin -2 147 483 648-den 2 147 483 647 diapazony bilen deňesdirin. signed tipli uýtgeýan ululykda belligi (nyşany) görkezmek üçin hyzmat edýan bir bit unsigned tipinde san ahmiýetiniň bölegi bolup durýar. Sonuň netijesinde insigned tipli uýtgeýan ululyklaryň oňyn ahmiýetleriniň uly diapazony bolýar, yöne ýaramaz ahmiýetleri görkezip bilmeýar. unsigned konstantasyna goşulan U (ýa-da u) harpyna üns beriň. Oňki mysalda men onuň long tipli konstantadygyny görkezmek üçin oňa L harpyny hem goşdum. Bu konstantalaryň modifikator simwollaryny islendik registrde we islendik yzygiderlilikde peýdalanyp bolar. Emma sular ýaly ähmiýetler görkezilende haýsydyr bir ýeke-tak stilden ugur alynsa gowy bolar.

unsigned esasy sözi tipin spesifikasiatory hökmünde aýratyn hem ulanyp bolar, mysal üçin, unsigned int.

Bellik, signed we unsigned sözleriniň esasy sözlerdigini ýatdan çykarmaň, şol sebäpli hem olary uýtgeýan ululyklaryň atlary hökmünde ulanmak bolmaz.

Logiki tip *Logiki uýtgeýan ululyklar* - bu dine iki ahmiýeti, ýagny true we false ähmiýetleri saklamaga ukyply uýtgeýan ululyklardyr. Sular ýaly logiki uýtgeýan ululyklar bul algebrasyny işlap düzen Jorj Bulyň ady boyunça bool atlandyrylýar. bool tipi bitin sanly tip hökmünde düşündirilyar. Logiki uýtgeýan ululyklar bul uýtgeýan ululyklar (Boolean) diýlip hem atlandyrylýar. bool tipli uýtgeýan ululyklar barlaglaryň netijesini saklamak üçin peýdalanylýar, olar ýa „hakykat“ (ture) ýa-da „ýalan“ (falše) ahmiýetlerini alyp biler, bir ahmiýetiň beýleki ahmiýete

dendigi barlananda şeýle edilýar, mysal üçin.

```
bool tipli uýtgeýan ululygyň ady şu operatoryň
kömegi bilen yglan edilýar
bool testResult;
```

Elbetde, bool tipli uýtgeýan ululygy olaryň yglan edilen wagtynda inisiiallaşdyryp bilersiniz.

```
bool colorIsRed = true;
```

ýa-da şunun ýaly:

```
bool colorIsRed (true);
```

Bellik. Kodda TRUE we FALSE ahmiýetlerine san tipleriniň ahmiýeti, hususan-da, int ýaly köp gabat gelip bolýar. Bu C++ dilinde bool tipiniň oranyşdyrylmadyk we logiki ähmiýetleri bermek üçin adatça int tipli uýtgeýan ululyklaryň ulanylýan döwründen galan mirasdyr. Şunda nol ahmiýetler „ýalan“ hökmünde, nol däl ahmiýetler bolsa „hakykat“ hökmünde düşündirilýardi. TRUE we FALSE simwollary MFC kitaphanasynda şu wagta çenli hem ulanylýar, bu ýerde olar degişlilikde, nol däl bitin ähmiýetleri we 0 ähmiýetleri görkezýärler. Baş harplar bilen ýazylan TRUE we FALSE literallarynyň C++ dilinde esasy söz bolup durmaýandygyna üns berin; bu dine MFC kitaphanasynda kesgitlenen simwollardyr. Şeýle hem TRUE we FALSE simwollarynyň bool tipli ýol berilýan ahmiýetler bolup durmaýandygyny ýatdan çykarmaň, sonuň üçin hem TRUE simwoly true bilen garyşdyrman.

Uýtgeýan oturly tipler

Bitin sanlara degişli bolmadyk sanly uýtgeýan ululyklar uýtgeýan oturly sanlar hökmünde saklanýar. *Uýtgeýan oturly* san 112,5 ýaly onluk ahmiýetli görnüşde ýa-da 1D25E2 ýaly eksponensial görnüşde (bu ýerde onluk bölegi E-den (eksponent) soň görkezilen derejede 10-a köpeldilýär) aňladylyp bilner. Şeýlelikde, soňky san $-1,125 \times 10^2$, bu 112,5-e dendir.

Bellik. Uýtgeýan oturly konstantalar onluk nokady, eksponenti ýa-da olaryň ikisini hem öz icine almalydyr. Eger siz san ahmiýetini olary peýdalanmazdan ýazýan bolsaňyz, onda bitin sanly ahmiýeti alýarsyňyz.

Uýtgeýan oturly uýtgeýan ululygy, aşakda görkezilýän operatordaky ýaly, double esasy sözi peýdalanmak bilen görkezip bilersiniz.

```
double in to__mm = 25.4;
```

double tipli uýtgeýan ululyk ýadyň 8 baýtyny eýeleýar we takyklygy 15 onlukly belgiler bilen kesgitlenýar. Olaryň diapazony 15 belgi bilen aňladardan has giňdir - $1,7 \times 10^{-308}$ we $1,7 \times 10^{308}$ çenli aralykda oňyn we yaramaz diapazonlar. Eger size double tipiniň üpjün edýan 15 belgili takyklyk we ahmiýetleriň örän uly diapazonlary gerek bolmasa, uýtgeýan oturly 4 baýty tutýan uýtgeýan ululyklary yglan etmek üçin float esasy sözi peýdalanyň.

```
float pi = 3.14159f;
```

Bu operator başlangyç ähmiýeti 3.14159 bolan pi uýtgeýan ululygy kesgitleýar. Konstantanyň ahyryndaky f simwoly onuň float tipligini görkezýär. f simwoly bolmasa konstantanyň double tipi bolardy. float tipi bilen yglan edilen

uýtgeýan ululyklaryň takmynan 7 onlukly belgiler ýaly takyklygy bolýar we $3,4 \times 10^{-3}$ -den $3,4 \times 10^3$ -e çenli oňyn we yaramaz ahmiýetlere ýol berýar.

C++ dili üçin ISO/IEC standart long double uýtgeýan oturly tipi hem kesgitleýar, ol Visual C++2010 gurşawynda double tipiniňki ýaly diapazon we takyklyk bilen ornaşdyrylandyr.

ISO/IEC C++ binýatlyk tipler

2.1 tablisada ISO/IEC C++ binýatlyk tiplerin ahlisiniň sanawy, şeýle hem olaryň Visual C++2010 gurşawynda ýol berilýan ahmiýetleri berilýar. **ISO/IEC C++ binýatlyk tipler**

4.1 tablisa

<i>Tipi</i>	<i>Baýtlard a ölçegi</i>	<i>Bahalaryň diapazony</i>
bool	1	true ýa-da false
char	1	Edil signed char ýaly: -128-den +127-ä çenli. Hökmany däl, erama char tipi üçin edil unsigned char tipiňki ýaly diapazony berip bilersiniz
signed char	1	-128-den +127-ä
unsigned char	1	0-dan 255-e çenli
wchar_t	2	0-dan 535-e çenli
short	2	-32 768-den +32 767-ä çenli
unsigned short	2	0-dan 65 535-e çenli
int	4	-2 147 483 648-den 2 147 483 647-ä çenli
unsigned int	4	0-dan 4 294 967 295-e çenli
long	4	-2 147 483 648-dan 2 147 483 647-ä çenli
unsigned long	4	0-dan 4 294 967 295-e çenli
float	4	$3.4 \times 10^+$, takmynan 7-razrýadly takyklyk
		$\pm 1.7 \times 10^{+30}$, takmynan 15-razrýadly takyklyk
long double	8	$\pm 1.7 \times 10^+$, takmynan 15-razrýadly takyklyk

Literallar

Şu kitapda uýtgeýan ululyklary inisiallaşdyrmak üçin örän köp açyk ahmiýetleri peýdalandyk. C++ dilinde islendik görnüşdäki kesgitli ähmiýetler literallar diýlip atlandyrylýar.

Literal - bu belli bir tipiň ahmiýetidir, sonun üçin 23, 3. 14159, 9. 5fwe true - bu deňşlilikde, int, double, float we bool tipli literallaryň mysallarydyr. "Samuel Beckett" literaly - bu setir literalynyň mysalydyr, ýöne bu tipi ara alyp maslahatlaşmagy indiki baba goýalyň. 4.2 tablisada dürli tiplerin literallarynyň ýazgylarynyň mysallary berildi.

Dürli tiplerin literallarynyň mysallary

4.2 tablisa

<i>Tipi</i>	<i>Literallaryň mysallary</i>
-------------	-------------------------------

char, signed char ýa-da unsigned char	•A', 'Z\ '8'. ...
wchar t	L'A', L'Z', L'8\ L'*
int	-77,65, 12345, 0x9FE
unsigned int	10U, 64000U
long	-77L, 65L, 12345L
unsigned long	5UL, 999999999UL, 25ul, 35U1
long long	-777LL, 66LL, 123456711
unsigned long long	55ULL, 999999999ULL, 885ull, 445U11
f l o a t	3.14f, 34.506f
double	1.414,2.71828
long double	1.414L, 2.71828L
bool	true, falše

Siz literalý shart ýa-da unsigned short tipine degişli hökmünde berip bilmersiňiz, yöne kompilyator, eger literalýň ähmiýeti uýtgeýan ululygyň tipi üçin ýol berilýan diapazonda bolýan bolsa, int tipli uýtgeýan ululyk üçin şol tipiň literallary bolup duran başlangyç ähmiýeti kabul eder.

Köplenç ýagdaýda size literallary mysal üçin, 12 fut ýaly ähmiýeti dýuýmlara ýa-da 25,4 dýuýmy millimetrlere öwürmek bilen, programmada hasaplamalaryň barşynda peýdalanmak gerek bolar. Emma, olaryň ahmiýetiniň aýdyň görnüp durmayan wagtynda programmalarda literallary açyk ulanmakdan gaça duruň. Siziň 2.545 ähmiýeti görkezyän wagtyňyzda onuň dýuýmdaky santimetrleriň sanyny aňladýandygy hemmelere hemişe aýdyň bolup durmaýar. Literala deň kesgitli ähmiýeti bolan uýtgeýan ululygy ygılan edeniň gowudyr. Mysal üçin, siz ony inçesToCentimeters (dýuýmlary santimetrlere) diýip atlandyryp bilersiniz. Sonda inçesToCentimeters uýtgeýan ululygy gabat gelende onuň manysy ýeterlik derejede aýdyň bolar. Şu bapda biraz sonra uýtgeýan ululygyň ähmiýetini nähili bellemelidigini gorersiňiz.

Maglumatlaryň tipleri **üçin** sinonimleri kesgitlemek

typedef esasy sözi maglumatlaryň bar bolan tipi üçin has ady kesgitlemäge mümkinçilik berýar. typedef esasy sözi peýdalanmak long int standart tipiniň ekwiwalenti hökmünde bilen BigOnes tiniň adyny kesgitläp bilersiniz, şunda operatorlar ulanylýar.

```
typedef long int BigOnes;
//Tipiň ady hökmünde BigOnes kesgitlemek
```

Bu BigOnes sözünü long int tipi üçin alternatiw at hökmünde kesgitleýar, soňa görä-de siz long int tipli mynum uýtgeýan ululygy şu görnüşde ygılan edip bilersiniz.

```
BigOnes mynum - OL;; //Tipiň ady hökmünde BigOnes
kesgitlemek
```

Hakykatdan, eger siz tipiň has adyny BigOnes ýaly kesgitleýan bolsaňyz, onda dürli uýtgeýan ululyklary yglan etmek üçin şol bir programmada iki spesifikatory hem ulanyp bilersiňyz, netijede olar kompilyátor tarapyndan bir tipli hökmünde kabul ediler.

typedef esasy sözün dine bar bolan tip üçin sinonimi kesgitleýandigi sebäpli bu artykmaç ýaly görnüp biler, ýöne bui beyle däl. Bu serişdaniň tipiň çylşyrymly spesifikasiýasy bolup duran bir ady kesgitlemegiň hasabyna çylşyrymly yglan etmeleri yönekeyleşdirmekde örän möhüm rol oýnaýandygyny geljekde gorersiňiz, bu siziň koduňyzy okamaga has ýaramly edýar.

Ahmiýetleriň belli bir toplumy **bolan** uýtgeýan ululyklar

Kähalatlarda sizde haýsydyr bir bellikler boyunca (mysal üçin, hepdäniň günleri, ýa-da ýylyň aýlary) salgylanylmagy amatly ýol berilýan ahmiýetleriň çakli toplumyndan ähmiýetleri kabul edip biljek uýtgeýan ululyklara zerurlyk yüze çykyp biler. C++ dilinde sular ýaly ýagdaýlar (icin yörite serişde göz önünde tutulandyr, ol sanap geçme (enumeration) diýlip atlandyrylýar. Görkezilen mysaly alalyň - hepdäniň günlerine laýyk gelyän ähmiýeti kabul edýän uýtgeýan ululyk. Onu şu görnüşde kesgitläp bolar.

```
enum Week{Mon, Tues, Wed, Thurs, Fri, Set, Sun}  
thisWeek;
```

Bu operator Week atly sanap geçme tipini we sanalyp geçilyän Week ululygynyň ekzemplyary bolup durýan thisWeek uýtgeýan ululygy yglan edýar, bu ululyk dine şekilli ýaýlaryň icinde yglan edilen konstant ähmiýetleri kabul edip bilýar. Eger siz thisWeek uýtgeýan ululyga görkezilen topara girmeýan zady berjek bolsaňyz, bu ýalňyslyga getirer. Bu ýaýyň icinde sanalyp geçilyän simwoliki atlar sanap geçýanler (enumerator) diýlip atlandyrylýar. Hakykatdan günlerin her bir ady kompilyátor tarapyndan awtomatiki usulda kesgitli bitin sanly ahmiýet hökmünde berilýar. Sanawdaky birinji at - Mon - 0 ähmiýeti alýar, Tues 1 bolar we ş.m.

Sanalyp geçilen konstantalaryň birini thisWeek uýtgeýan ululygyň ähmiýeti hökmünde berip bilersiniz.

```
thisWeek = Thurs;
```

Sanalyp geçilen konstantalary sanap geçme ady bilen kwalifikasiýa bölmegin gerek dældigine üns beriň. thisWeek uýtgeýan ululygyň ähmiýeti 3-e deňdir, sebäbi sanap geçmede kesgitlenen simwoliki konstantalara duýdurmazdan, O-dan başlap ähmiýet berilýar.

Dudurmazdan, soňky sanap geçmelerin her biriniň ähmiýeti önküdakiden bir birlik uludyr, ýöne siz 0-dan tapawutlanýan başlangyç ähmiýeti açyk görkezmek isleseňiz, onda şeýle ýazyň.

```
enum Week{Mon = 1, Tues, Wed, Thurs, Fri, Set, Sun}  
thisWeek;
```

Indi sanap geçmedeki konstantalaryň belgileri 1-den 7-i aralygynda bolar. Sanap geçýanleriň unikal ahmiýetiniň bolmagy hökman däl. Siz Mon hem, Tues

hem , mysal üçin, 1 ahmiýetli bolmaly diýip kesgitlep bilersiniz, bu şu operatioriaryň kömegi bilen edilyär.

```
enum Week{Mon - 1, Tues - 2, Wed, Thurs, Fri, Set, Sun} thisWeek;
```

Hakykatdan thisWeek uýtgeýan ululygyň tipi int bolup durýar, ol sanalyp geçilen tipiň ähli uýtgeýan ululyklary ýaly 4 baýty tutar.

Sanap geçyanleri inisiallaşdyrmak üçin funksiýalaryň notasiýalaryny peýdalanmaga rugsat edilmeyändigini ýatdan çykarmaň. Ozal getirilen mysallardaky ýaly, ahmiýet berme operatoryny peýdalanmalydyr.

Sanap geçmaniň kesgitlemesi bar wagtynda şol ötipli beýleki uýtgap durýan ululygy kesgitleň.

```
enum Week nextWeek;
```

Bu nextWeek uýtgeýan ululygy ozal görkezilen ähmiýeti alyp bilýan sanap geçme hökmünde kesgitleýar. Uýtgeýan ululyk yglan edilende enem esasy sözi hem geçirip bilersiniz, şonuş üçin önki operator şunun ýaly ýazylyp bilner.

```
Week nextWeek;
```

işleginize göre ähli sanap geçijilere belli bir ähmiýetleri berip bilersiniz. Mysal üçin, şu hili sanap geçmani kesgitlep bolar.

```
enum Punctuation {Comma = Exclamation = Question =? things;
```

Bu ýerde siz things uýtgeýan ululygy degişli simwollaryň san ekwiwalentleri hökmünde kesgitleýarsıñiz. Bu simwollaryň kodlary onluk görnüşinde, degişlilikde, 44, 33 we 63 bolup durýar. Görşüniz ýaly, sanap geçijilerin ahmiýetiniň ýokarlanýan tertipde gitmegi hökman dälidir. Eger siz ähli ähmiýetleri aýyk görkezmeyän bolsañyz, onda ähmiýeti berilmedik her bir sanap geçijä, ikinji mysalymyzdaky Week sanap geçmede bolşy ýaly, ozalka garanynda bir birlik uly ähmiýet berler.

Eger sonra sanalyp geçilýan tipiň beýleki uýtgeýan ululyklaryfny yglan etmek zerurlygy ýoki boldsa, bu tipiň adyfny geçirin, mysal üçin.

```
enumWeek{Mon,Tues,Wed,Thurs,Fri,Set,Sun}thisWeek,nextWeek, J astWeek;
```

Bu ýerde Mon-dan başlap Sun-a çenli ähmiýetleri kabul edip biljek üç uýtgeýan ululyk yglan edilendir. Sanap geçijiniň tipiniň görkezilmändigi sebäpli, siz oňa sonundan salgylanyp bilersıñiz. Siz umuman bu sanap geçmaniň tipiniň beýleki uýtgeýan ululyklaryny yglan edip bilersıñiz, sebäbi bu kesgitlemäni gaýtalamaga hem rugsat berilmeyär. Şeýle etmäge synanyşyk sizñ Mon-dan başlap Sun-a çenli ähmiýetleri gaýtadan kesgitlemekçi bolýandygyñyzy aňladardy, muňa bolsa ýol berilmeyär.

Binýatlyk giriş-çykyş amallary

Bu ýerde siz C++ dilinde giriş we çykyş bilen, dil mundan beýläk öwrenilende ony mysallarda peýdalanmak üçin ýeterlik bolan derejede ýüzleý tanyşarsyňyz. Bu kyn däldir, tersine örän ýeňildir. Windows amal ulgamy üçin programmirlemede bolsa ol asla gerek hem bolmaz. C++ dilinde giriş we çykyş maglumat akymalary düşünjesiniň daşynda aýlanýar; maglumatlary çykyş akymyna goşmak we giriş akymyndan kabul etmek. Siz ekranda buýruk setiriniň ISO/IEC C++ standart çykyş akymynyň cout diýlip atlandyrylýandygyny gördünüz (console output konsola çykyş). Klawiaturadan onun üstüni ýetirýän giriş akymy cin (console input - konsoldan giriş) ady bilen bellidir.

Klawiaturadan giriş

Siz klawiaturadan girişi akymdan çykarmak operatoryny » peýdalanmak bilen, cin akymynyň üstünden alýarsyňyz. num1 we num2 uýtgeýan ululyklara klawiaturadan okamak üçin şu operatory ýazyp bolar.

```
si:d: :cin >> num1 >> num2 ;
```

Çykarmak operatory (») maglumatlaryň berilýan ugruny, häzirki ýagdaýda cin-den nobat boýunça uýtgeýan iki ululuga berilýändigini görkezyär. Önünden gelyän ähli tesimler geçirilýar we klawiaturadan girizilen birinji bitin ahmiýet num1 uýtgeýan ululyga gelip gowusýar. Bu giriş operatorynyň çepden saga ýerine ýetirilýandigi sebäpli bolup geçýar.num1 uýtgeýan ululygyň sonundan gelyän islendik tesimler inkâr edilýar we ikinji girizilen bitin sanly ahmiýet num2 uýtgeýan ululyga okalýar. Biri-biriniň yzyndan gelyän beýleki ahmiýetleriň arasynda olary bölmek üçin haýsydyr bir probel simwollar bolmalydyr. <Enter> klawisine basan wagtyňyzda giriş akymy amaly tamamlanýar we programmany ýerine ýetirmek nobatdaky operatorlardan başlanyar. Elbetde, nädogry maglumatlar girizilende ýalňyslar yüze çykyp biler, ýöne men siz hemişe dogry ýerine ýetirersiňiz diýip pikir edýarin.

Uýtgeýan oturly ahmiýet edil bitin sanlar ýaly, klawiatura girişinden okalýar, şunda siziň olary garysdyrmagyňyz mümkindir. Akymlaýyn giriş we onuň amallary islendik binýatlyk tipiň uýtgeýan ululyklary we maglumatlary awtomatiki usulda tanaýar. Şu operatorlara seredeliň.

```
int num1 = 0, num2 = 0;  
double façtor = 0.0;  
cin >> num1 >> façtor >> num2;
```

Bu ýerde soňky setir bitin ahmiýeti num1 uýtgeýan ululyga, soňra bolsa uýtgeýan oturly ahmiýeti façtor uýtgeýan ululyga okaýar, iň soňunda ýene bir bitin sanly ahmiýeti num2 uýtgeýan ululyga okaýar.

Buýruk setirine çykyş

Yokarda seredilip geçilen mysallarda siz buýruk setirine çykyşy gördünüz, men şonda-da bu tema gaýdyp gelesim gelýar. Maglumatlary displeýe çykarmak girişin üstüni ýetirýan usul bilen amala aşyrylyar. Siziň eýýam görüp geçişiniz ýaly, çykyş akymy cout diýlip atlandyrylýar we oňa maglumatlary geçirmek üçin giriş operatory « peýdalanýlýar. Bu operator maglumatlaryň hereketiniň ugruny hem „görkezyär". Siz ony ozal dyrnak icine alnan tekst setirlerini çykarmak üçin ulanypdyňyz. Men uýtgeýan ululygyň ahmiýetini çykarmak prosesini görkezyän.

Amaly ders

Buýruk sertirini çykarmak

Siz eýýam boş taslama döredipsiniz, oňa başlangyç kodun täze faylyny goşupsynyz we ýerine ýeti rilýan programmany ýazypsyňyz diýeliň. Aşakda Ex2 02 taslamasy döredilenden soň başlangyç koduň faýlyna ýerlesdirilmeli kod görkezilyär.

```
// Ex2    02 . cpp
// Çykarmak boýunça yumuş
#include <iostream>
using std::cout;
using std::endl;
int main ( )
{
    int num1 = 12 34, num2 = 5678;
    cout << endl;           // Taze setiri başlamak
    cout << num1 << num 2 // iki uýtgeýan ululygy çykarmak
    cout << endl; // Setiri tamamlamak
    return 0;               // Programmany tamamlamak
}
```

Ex2_02. cpp kodunyň fragmenti

Bu nähili isleyär

Koduň başynda std: :cout we std: :endl atlary üçin using yglan etmegin bardygy sebäpli, olary kodda atlar ginişligingi görkezmezden peýdalanyň. main() funksiýasynyň bedenindäki birinji operator num1 we num2 bitin sanly iki uýtgeýan ululygy yglan edyär we inisiallaşdyryar. Ondan soň iki çykyş operatory gelýar, olaryň birinji si ekrandaky kursoryň ornuny täze setire geçirýar. Çykyş operatorlarynyň çepden saga ýerine ýetirilýandigi sebäpli, ikinji operator num1 uýtgeýan ululygyň ähmiýetini görkezyär, onuň yzyndan num2 ähmiýeti gelýar.

Yokarda getirilen kódy kompilirläp, işe goýberen wagtyňyzda ekranda şu çykyşy alarsyňyz.

12345678

Çykyş dogry, yöne beýle bir peýdaly däl. Hakykatdan siz iň bolmanda bir probel bilen bölünen iki aýratyn ähmiýeti görmelisiniz. Duýdurmazdan, çykyş akymy ähmiýeti görkezyän onluk sanlary görkezyär, bu bolsa çykarylyan ahmiýetleriň arasyny haýsydyr bir probel bilen bölmegi göz önünde tutmaýar. Sonun üçin hem bir ahmiýetiň haçan tamamlanýanyp, beýlekisiniň bolsa haçan başlanyandygyny aýtmaga siziň hiç hili mümkinçiliginiz ýokdur.

Çykyşy formatirlemek

Ahmiýetleriň arasynda probelleriň ýokdugy bilen bagly problemany çykyş akymynda ahmiýetleriň ikisiniň arasyna probel goymak bilen, örän ýeňil düzedip bolar. Muny ilki başky programmada şu setiri çalyşmak bilen edip bolar.

```
cout << num1. << num2 // İki ähmiýeti çykarmak
şu setir bilen
cout << num1 << << num2; // iki ähmiýeti çykarmak
```

Elbetde, eger sizde birnäçe çykyş setiri bar bolsa we siz kolonkalary deňlemek islenyän bolsaňyz haýsydyr bir goşmaça mümkinçilikler gerek bolar, çünki siz her bir ähmiýetde näçç alamatyň boljakdygyny bilmeýarsyňiz. Bu ýerde manipulýatorlar diýilýanlar kömek eder. Manipulýator maglumatlaryň akyma çykarylyşyny (ýa-da girişden çykarylyşy) dolandyrmagyň usulyňy uýtgedýar.

Manipulýatorlar iomanip sözbaşysynyn faýlynda kesgitlenýar, şol sebäpli onuň üçin hem #include direktiwasyňy goşmaly bolar. setw (n) manipulýatory gerek bolar. Ol ini n bolan meýdanda probelleri deňlemek bilen soňky ähmiýetleri çykaryar, ýagny setw (6) manipulýatory ini 6 simwol bolan meýdana onuň sonundan gelyän ähmiýeti berer 7.

Muňa jikme-jik seredeliň.

Amaly ders

Manipulýatory ulanmak

Özüňize gerek bolan çykyşa has menzeş bolan çykyşy almak üçin size programmany şu görnüşde üýtgetmek gerek bolar.

```
// Ex2_03. cpp // Çykarmak boyunca yumuş #include
<iostream> #include <iomanip>
```

```
using      std:
:cout;     using
std:        :endl;
using std: :s e
t w;

int main()
{
int num1 = 1234, num2 = 5 67 8;
cout << endl; // Täze setiri başlamak
cout << setw (6) << num1 << setw (6) << num 2
    // iki uýtgeýan ululygy çykarmak
cout << endl; // Setiri
tamamlamak
return 0; // Programmany tamamlamak
}
```

Ex2_03. cpp kodunyň fragmenti

Bu nähili işleyiir

Soňky mysalda girizilen uýtgetmeleriň arasynda - iomanip sözbaşysy üçin #include direktiwasy goşulypdyr, std atlar giňisliginden setw ady üçin using yglan etme we her bir uýtgeýan ululygyň ähmiýetlerini çykarmazdan ön çykyş akymyna setw () manipulýatory goşulypdyr, şeýlelikde olaryň ähmiýeti ini 6 simwol bolan meýdana çykarylyar. Netijede iki ähmiýet arasy bölünen takyk çykyş alarsyňyz:

12 3 4 5 67 8

Setw () manipulýatorynyň dine ýeke-tak çykyş ähmiýeti bilen işleyändigine üns beriň, ol akyma gosundynyň gönüden-göni yzyndan gelýar. Siz manipulýatorlary belli bir indäki meýdana deňlemek isleýan her bir ähmiýetiňiziň

gönüden-göni önünde goýmalysyňyz. Dine bir stew () manipulýatory goýýan bolsaňyz, ol dine onuň yzyndan iberilen birinji ahmiýete täsir eder. Soňky islendik ahmiýetler adaty görnüşde çykarylýar. Iň soňky mysaldaky ikinji setw (6) manipulýatory we onuň goýlus operatoryny aýyrmak bilen muňa göz ýetirip bilersiniz.

Dolandyrýan yzygiderlilik

Goşa dyrnak icine alnan simwol setirini ýazýan wagtyňyzda oňa ýorite simwollary goşun bilersiniz, olar *dolandyrýan simwollar* (escape sequence) diýlip atlandyrylýar. Simwollaryň adaty interpretasiýasyny ýatyrýandyklarynyň (escaping) hasabyna başgaça görkezilip bilinmejek simwollary setire ýerleşdirmäge mümkinçilik berýandigi üçin olar şeýle atlandyrylýar. Dolandyrýan yzygiderlilik tersine ýapgyt çyzykdan (\) baslanýar, ol kompilýatory özüniň yzyndan gelýan simwoly aýratyn obraz ýaly interpretasiýa etmäge mejbur edýar. Mysal üçin, tabulýasiýanyň simwoly \t ýaly ýazylýar, şeýlelikde t simwoly setirde kompilýator tarapyndan t harp ýaly däl-de tabulýasiýanyň simwoly hökmünde kabul edilýar. iki çykyş operatoryna serediň. cout << endl << "Bu çykyş.";

```
cout << endl << "\Bu tabulýasiýadan soňky çykyş.";
```

Olar ekrana şu setirleri çykararlar:

Bu çykyş.

Bu tabulýasiýadan soňky çykyş.

\t yzygiderliliği ikinji çykyş operatorynda özüniň yzyndan barýan teksti tabulýasiýanyň birinji ornuna süýşürýar.

Hakykatdan endl esasy sözi peýdalanmagyň ýerine her bir setirde täze setiriň \n dolandyrýy yzygiderlilik simwoly ulanyp bilersiniz, sonun üçin hem ondan operatoriary şu görnüşde gaýtadan ýazyp bol ar:

```
cout << "\nBu çykyş.";
```

```
cout << "\n\tBu tabulýasiýadan soňky çykyş.";
```

4.3 tablisada gerek bolmagy mümkin kabir dolandyrýan yzygiderlilik sanalyp geçilyar.

Kabir peýdaly dolandyrýan yzygiderlilikler

4.3 tablisa

<i>Dolandyrýan yzygidelik</i>	<i>Nâme edýar</i>
\a	Ses arkaly signal (duýduryş)
\n	Täze setiriň simwoly
\v	Bir dyrnak
\w	Tersine ýapgyt çyzyk
\b	Bir simwol yza süýşmek
\t	Tabulýisiýanyň simwoly
\"	Goşa dyrnak
\?	Sorag alamaty

Eger siz tersine ýapgyt çyzygy ýa-da goşa dyrnagy setire goşmak isleseňiz, onda olary görkezmek üçin degişli dolandyryýan yzygiderlilik peýdalanyň. Tersine bolanda tersine ýapgyt çyzyk beýleki dolandyryýan yzygiderliliğiň başy hökmünde, goşa dyrnak bolsa - simwol setiriniň ahyry hökmünde interpretirlener. Şeýle hem siz char tipli uýtgeýan ululyklar inisiallaşdyrylanda dolandyryýan yzygiderlilikler tarapyndan kesgitleýän simwollary ulanyp bilersiniz.

char Tab -^~\t; //Tabulýasiýa simwoly bilen inisiallaşdyrmak Simwoll literalynyň bir dyrnak alamaty bilen çäklenýändigini sebäpli, bir dyrnagynyň özüni, ýagny 'V ybarat simwol literalyny görkezmek üçin dolandyryýan yzygiderlilik peýdalanyň.

Amaly ders Dolandyryýan yzygiderlilikleri peýdalanmak
Aşakda önki tablisadaky kabir dolandyryýan yzygiderlilik peýdalanýan programmanyň teksti getirilýär.

```
// Ex2 04П .cpp
//Dolandyryýan yzygiderlilikleri peýdalanmak
#include <iostream>
include <iomanip> using std: : cout; int main()
{
    char newiine - '\r+; // Dolandyryjy yzygiderlilik -
täze setiriň simwoly
    cout << newiine: //Täze setiri başlamak
    cout << „\ „WeV ll make our esçapers in sequeñçe“,
he said.“;
    cout << „\n\tThe programWs över, itVs time take
make a beep beep. \a\a“;
    cout << newiine; // Täze setiri başlamak
    return 0; // Programmadan çykamak
} Exl 02A . cpp
kodunyň fragmenti
```

Eger siz bu programmany kompilirleseňiz we işe goýberseňiz, onda şu çykyşy görersiniz.

```
"We"ll make our esçapers in sequeñçe", he said.
The program\ " s över, itVs time take make a beep
beep.
```

Bu nähili işleyär?

main() funksiýasynyň birinji setiri newiine uýtgeýan ululygy kesgitleýär we ony täze setiriň simwoly bilen inisiallaşdyrýar. Soňra standart kitaphanadaky endi esasy sözün ýerine newiine uýtgeýan ululygy ulanyp bilersiniz.

newiine uýtgeýan ululygy çykyşa ugradylandan soň goşa dymagyny (V) we bir dyrnagyny (V) simwollaryny görkezmek üçin dolandyryş yzygiderliliğini ulanýan setir emele gelýär. Size bir dyrnak üçin dolandyryýan yzygiderlilik peýdalanmak hökmany däl, sebäbi setir goşa dyrnaklar bilen çäklendirilendir we kompilyator ondaky bir dyrnagyny bölüji hökmünde däl-de, bolşy ýaly kabul edýär.

Элита гоша дырныг görkezmek üçin bu setirde dolandyryňan yzygiderlilik ulanmak zerurdyr. Setir dolandyryňan yzygiderlilikden - täze setiriň simwolyndan başlanyar, onuň yzyndan tabulýasiýa simwolyň dolandyryjy yzygiderlilik gidyär, sonun üçin hem çykyş setiri tabulýasiýanyň omuna saga süýşyar. Setir hem ses arkaly duýduryş berýan yzygiderlilikleriň iki ekzemplýary bilen tamamlanýar, sonun üçin hem siz kompýuteriň dinamiginden bassyr iki ses duýduryşyny eşidersiniz.

4.6. C++ dilinde hasaplamalar

Bu ýerde siz hakykatdan hem girizilen maglumatlar bilen nâmedir bir zatlary edip başlarsynyz. Siziň ýonekeý giriş-çykyş bilýandigiňiz üçin, aralykdaky bölege, ýagny C++ programmasynyň maglumatlary işlap geçmek bilen meşgullanyan bölegine ýuzleneliň. C++ diliniň hasaplamalar bilen bagly ähli aspektleri intuitiw derejede ýeterlik düşnükli, diýmek, bu temany öwrenmek bize örän ýenil bolar. **Ahmiýet bermek** operatory

Siz ahmiýet bermek operatorlarynyň mysallaryny eýýam gördünüz. Tipiki at bermek operatory şunun ýaly görünyar.

```
whole = part1 + part2 + part3;
```

Ahmiýet bermek operatory deňlik alamatyndan soň sagda durýan aňlatmanyň ahmiýetini - şu ýagdaýda part1, part2 we part3 uýtgeýanleriň ahmiýetleriniň jemini hasaplamaga, soňra bolsa, deňlik alamatynyň çepinde görkezilen - häzirki ýagdaýda whole uýtgeýande netijäni saklamaga mümkinçilik berýar (bu ýerde whole - yöne bölekleriň jeminden başga zat däldir).

Bellik. Bu operatoryň, hemişe bolşy ýaly, nokatly otur bilen tamamlanýandygyna üns berin.

Siz gaýtalanýan ahmiýet bermeleri hem ýazyp bilersiniz.

```
a = b = 2;
```

Bu b uýtgeýan 2 ahmiýet bermek, sonundan b uýtgeýan ululygyň ahmiýetini a uýtgeýan ululyga bermek bilen menzeşdir (ekwiwalentdir). Arifmetiki **amallar**

Siziň ygtyýaryňyzda bolan binýatlyk arifmetiki amallar - goşmak, aýyrmak, köpeltmek we bölmek - degişlilikde, +, -, * we / simwollary bilen aňladylýar. Olar esasan, garaşlyşy ýaly isleýarler, dine bölmek kadadan çykyar, olar bitin sanlar ýa-da konstantalar babatynda ulanylanda onuň özüni alyp barşynda (siz ony görersiniz) kabir gysarmalar bardyr. Şunun ýaly operatory ýazyp bilersiniz.

```
netPay = hours * rate - deductions;
```

Bu ýerde hours we rate uýtgeýanleriň ahmiýetleriniň köpeltme hasyly hasaplanyp çykarylýar, soňra bolsa alnan netijeden deductions uýtgeýan ululygyň ähmiýeti aýrylar. Köpeltmek we aýyrmak amallary goşmak we aýyrmak amallarynyň önünden ýerine ýetirilýar, ol şeýle hem bolmalydy. Şu bapda men anlatmalarda dürli amallary ýerine ýetirmegiň tertibi barada giňişleýin gürrün bererin. hours * rate - deductions aňlatmasyny hasaplamagyň umumy netijesi netPay uýtgeýan ululykda ýerleşýar.

Soňky operatora peýdalanylan „aýyrmak“ („minus“) alamaty iki operandlar bilen iş salyşyar - ol çepdäki operandyň ahmiýetinden sagdaky operandyň

ahmiýetini aýyrýar. Ol *binar amaly* diýlip atlandyrylýar, sebäbi bu ýerdi iki ahmiýet bardyr. „Aýyrmak“ alamaty hem bir operand bilen peýdalanyp bilner - şunda ol operandyň alamatyny tersine alamat bilen çalyşýar. Şu ýagdaýda ony *imar amal* diýip atlandyryrlar. Şu kódy ýazyp bilersiniz.

```
int a = 0;
int b = -5;
a = -b; // Aýyrmak alamaty operandyň alamatyny
uýtgedýar
```

Bu ýerde a uýtgeýan ululygak +5 ahmiýet berler, sebäbi unar minusy b operandyň alamatyny uýtgçtdi.

Ahmiýet bermek döp bolan algebrada kabul edilişi ýaly, deňlige ekwiwalent dälidir. Ol faktyň konstataciýasyny däl-de, ýerine ýetirilmeli hereketi kesgitleýar. Ahmiýet beryän operatorndan sagdaky aňlatma hasaplanyp çykarylýar we netije çepdaky uýtgeýande saklanýar.

Bellik. Düzgün bolşy ýaly, ahmiýet beryän operatorndan çepdaky anlatma - uýtgeýan ululygyň bir adydyr, bu hökmany dälidir. Bu käbir görnüşin aňlatmasy bolup biler, eger bu anlatma bolsa, onda ony ýerine ýetirmegin netijesi 1-ahmiýet bolmalydyr (Ivalue). L-ahmiýet ýadyň hemişelik fragmenti bolup durýar, bu ýerde ähmiýet beryän operatorndan sagdaky aňlatmanyň netijesi saklanyp biler.

Indiki operatora seredeliň.

```
number = number + 1;
```

Bu "number uýtgeýaniň şu wagtky ähmiýetine birligi goşmaly, soňra bolsa netijäni gaýdymlaýyn number uýtgeýande saklamaly" diýmegi aňladýar. Eger oňa algebra aňlatmasy hökmünde seretseň, onda onuň manysy ýokdur.

Amaly ders

Binýatlyk arifmetikadaky gönükmeler

Siz bir otaga ýelmäp çykamak üçin näçe sany standart rulonlaryň gerek boljakdygyny hasaplamak bilen, C++ dilinde arifmetikada tejribe geçip bilersiniz. indiki mysa! ony ýerine ýetirýar.

```
// Ex2 05A .cpp
// Otagyň diwary üçin zerur bolan oboý rulonlarynyň
sanyny hasaplamak
#include <iostream>
using std : : count;
using std : : cin;
using std : : endl;
int main()

double height = 0. 0, width =0. 0, length = 0. 0;
// Otagyň ölçegleri
double perimeter = 0.0;
//Otagyň perimetri
const double rollWindth = 21.0;
//Standart rulonyň ini
const double rollLength = 12.0*33.0;
//Standart rulonyň uzynlygy
// (33 fut)
```

```

int strips per-roll = 0;
//Rulonda zolaklaryň sany
int strips^reqd = 0;
//Zolaklaryň gerek bolan sany
int nroll s = 0;
//Rulonlaryň umumy sany
cout << endl
//Täze set iri başlamak
cout << „Dýuýmda otagyň beýikligini giriziň: ";
cin >> height;
cout << endl;
//Täze setiri başlamak
cout << „Dýuýmda uzynlygyny we inini giriziň: ";
cin >> lenght >> width;
strips per roli = rollLength / heght;
//Rulondaky zolaklaryň sanyny almak
perimetr = 2.0* (length + width
//Otagyň perimetrini hasaplap çykarmak
strips reqd = perimetr / rollwidth;
//Rulonlaryň zerur bolan sanyny almak

nrolls = strips reqd / strips per roll;
//Rulonlaryň sanyny hasaplap çykarraak
cout<<endl<< „Siziň otagyňyza oboý ýelmap çykamak
üçin"
<< nrolls << „rulon gerek bolar." <<endl;
return 0;
}

```

Ex2_05A .cpp koduň fragmenti

Eger siz klawiaturada ýazmakda men ýaly ökde däl bolsaňyz, onda kompilyasiýa etmek babatynda birinji synanyşykda käbir ýalňyslaryň bolmagy ähtimaldyr. Siziň olary düzeden halatyňyzda, programma kompilyasiýa edilmelidir vve üstünlikli işlemelidir. Siziň kompilyatorda bir-iki gezek duýdurys habaryny almagyňyz mümkin. Yöne olary alada etmän - kompilyator dine size öz edýan işinize düşünmäge mümkinçilik beryär. Yalňyslyklar baradaky habarlaryň sebäplerini men soňra düşündirerin.

Bu n äh ili iş ley är?

Ilkibaşda bir zady belläsim gelyär - hasaplanyňyzda şu programmadan peýdalanan bolsaňyz artyk harçlan oboýyňyz üçin men jogapkarçilik çekyan däldirin. Zerur bolan rulonlar hasaplananda bu programmanyň berýan ýalňyslyklary C++ diliniň nähili isleyändigini sebäpli bolup geçyar we otaga oboý ýelmenende zaýаланan möçberi 50%-den gowrak bolup biler. Men bu mysalyň ähli operatorlaryny birin-birin barlap geçerin, gyzykly, özboluşly, hatda özüne çekiji pursatlary belläp geçerin, main() funksiýasynyň bedeninden soň gelýan operatorlar - siz üçin önden mâlim meydan bolup durýar, men hem şundan ugur alaryn.

Programmanyň kompozisiýasy bilen bagly bir-iki sany möhüm pursatlary belläp geçmek gerek. Birinjiden, operatorlar main() funksiýasynyň bedeninde ýeňil okap bolar ýaly deňeçerlenendir we ikinjiden, operatorlaryň dürli toparlary,

funksional toparlary bölüp aýrar ýaly, boş setirler bölen bölünendir. Operatorlaryň tesimleri - bu C++ programma koduny guramagyň fundamental tehnikasydyr. Siz programmanyň dürli logiki bloklaryny wizual bölmegi üpjün etmek üçin onuň ähli ýerde ulanylýandygyny gorersiňiz.

const modifikatory

main() funksiýasynyň bedeninde ilki başdan programmanyň peýdalanýan uýtgeýanleriniň bildirişler blogy bolýar. Bu operatorlar size tanyşdyr, ýöne olaryň arasynda öz icine haýsydyr bir tazeligi alýan iki operator bar:

```
const double rollWindth = 21.0;           //Rulonyň standart ini
const double rollLength = 12.0 * 33.0;    //Rulonyň standart uzynlygy
                                           // (33 fut)
```

Olaryň ikisi hem const diýen täze esasy sözden başlanyar. Bu tipin modifikatory bolup, uýtgeýan ululygynyň dine bir double tipi bolman, eýsem konstanta hem bolup durýandygyny görkezýär. Siziň kompilýatora bu iki uýtgeýan ululygynyň - konstanta bolup durýandygy barada anyk habar berýändiginiň sebäpli, ol bu uýtgeýan ululyklaryň ahmiýetini uýtgetmage synanysýan islendik operatory barlap biler we eger şunun ýaly ýagdaýy ýüze çykarsa ýalňyslyk barada habar berer. const hökmünde ygylan edilen uýtgeýan ululyk 1-ahmiýet bolup durmaýar, sonuň üçin hem ahmiýet beryän operatoryň çepinde ýerleşdiriler.

Siň programmanyň koduna konstanta ygylan edilenden soň rollwidth goşup muňa göz ýerip bilersiniz, şunun ýaly operator.

```
rollwidth = 0;
```

Şundan soň programma kompilýasiýany bes eder we ýalňyslyk barada habar berer 'error C3892: ' rollWiduy : you cannot assign to a variable that is const' (ýalňyslyk C3892: 'rollWidth': konstanta bolup durýana uýtgeýan ululygynyň ahmiýetini bermek bolmaz.

Kähalatda uýtgeýan ululyk tipleri const modifikatory bilen üpjün edip, programmada peýdalanylýan konstantalary kesgitlemek örän amatly bolup durýar, hususan-da käbir konstantalary programmada birnäçe gezek peýdalanýan wagtyňyzda. Birinjiden, bu bellenen maksady näbelli bolan literallary tutuş programma dargadandan has gowudyr. Mysal üçin, programmadaky 42 ahmiýeti köp zady aňladyp biler, ýöne siz 42 ahmiýeti bolan myAge (meniň ýasym) konstantany peýdalansaňyz, onda onuň ahmiýeti dessine aýdyň bolar. Mundan başga-da, eger size ulanylýan konstantanyň ahmiýetini uýtgetmek gerek bolsa, ony dine bir gezek, başlangyç kodun bir faýlynda etmeli bolarsyňyz, bu agzalan uýtgeýan ululygyny peýdalanylýan ähli ýerinde şu uýtgeýan ululygyny peýda bolmagyny kepillendirmek üçin şeýle edilýar. Siz bu tehnikany örän ýygý peýdalanarsyňyz.

Hemişelik ululykly (Konstant) anlatmalar

rollLength konstantasy hem arifmetiki anlatmalar bilen inisiallasdyrylýar (12.0*33.0). Uýtgeýan ululyklary inisiallaşdyrmak üçin konstanta aňlatmalary peýdalanmak mümkinçiligi programma ýazylanda olary el bilen hasaplamak zerurlygyndan halas edýar. Onsoňam sular ýaly anlatmalar köp maglumat berip biler, sebäbi, mysal üçin, ýöne 396 diýip ýazylanyna garanynda her biri 12 dýuým

bolan 33 fut diýlende ululygyň manysy has aýdyň görünyar. Adatça kompilyator konstant aňlatmalary takyk hasaplap çykaryar, şol bir wagtyň özünde ony el bilen etmägeç synanyşyk, aňlatmanyň çylşyrymlylygyna we siziň hasaplama ukybyňyza baglylykda, ýalňyslyga getirip biler.

Siz hasaplamalaryň netijesinde kompilyasiýa wagtynda konstanta ahmiýetiň berýan islendik aňlatmasyny, ýokarda kesgitlenen konstanta obýektleri goşmak bilen, peýdalanyň bilersiniz. Sonuň üçin hem oboýlaryň standart rulonynyň meýdanyny şunun ýaly ygylan ediň:

```
const double rollArea = rollWidth*rollLength;
```

Bu operator rollArea kontantasy inisiallaşdyrylanda peýdalanylan iki konstanta ygylan edilenden soň ýerlesdirilmelidir, sebäbi konstanta anlatmada peýda bolýan ähli anlatmalar kompilyatora başlangyç koduň konstanta anlatma peýda bolýan nokadynda kompilyatora mälüm bolmalydyr. *Programmány girizmek*

Kabir bitin sanlardan durýan uýtgeýan ululyklary ygylan edilenden soň programmanyň aşakda görkezilyän dört operatory klawiaturadan girişi işlap geçýärler.

```
cout << endl                                //Taze setiri başlamak
<< „Dýuýmda otagyň beýikligini giriziň:
"; cin >> height;
```

```
cout << endl                                //Taze setiri
başlamak
<< „Dýuýmda uzynlygyny we inini
giriziň: "; cin >> lenght >> width;
```

Bu ýerde konsola tekst girizilýar, bu tekst peýdalanyjyny cin standart giriş akymyny peýdalanmak bilen klawiaturadan zerur maglumaty girizmäge çagyryar. Siz ilkibada height uýtgeýan ululygyň ähmiýetini alýarsyňyz, soňra bolsa length we width uýtgeýan ululyklaryň ähmiýetini okaýarsyňyz. Hakyky programmada size girizen maglumatlaryňyzy mümkin bolan ýalňyslar babatynda barlamaly we okan ahmiýetleriňiziň manysynyň bardygyna göz ýetirmeli bolarsyňyz, yöne hazirlikçe munuň üçin siziň biliminiz ýeterlik dälir!

Netijäni hasaplamak

Seredilyän programmada dört operator bar, olar otagyň diwaryna ýelmemek üçin zerur bolan standart oboý rulonlarynyň sanyny hasaplamakda peýdalanylýar:

```
strips per ^roll = rollLength/heght;
//Ru'iondaky zolaklaryň sanyny
 almak perimetr - 2.0* (length +
 width //Otagyň perimetrini
 hasaplap çykarmak strips reqd -
 perimetr / roliwidth;
//Rulonlaryň zerur bolan sanyny
 almak nrolls = strips reqd /
 strips per roll; //Rulonlaryň
 sanyny hasaplap çykarmak
```

Birinji operator rulon otagyň beýikligine bölünende bir standart rulon kesilende oboýlaryň uzynlygy otagyň beýikligine deň bolan zolaklarynyň sanyny hasaplap çykaryar. Başga sözler bilen aýdylanda, eger otagyň beýikligi 8 fut bolsa, onda siz 396-ny 96 bolýarsıňız, ol uýtgeýan otur bilen, 4,125-ä deň bolan netijäni berer. Yöne bu ýerde bir inçe mesele bar. Siziň netjäni ýerlesdirýan uýtgeýan ululygyňyz - `strips__per_jroll - int` hökmünde yglan edilipdi, sonuň üçin hem ol dine bitin san ähmiýetini saklap bilyär. Şol sebäpli uýtgeýan oturly islendik ahmiýeti bitin san ähmiýeti hökmünde saklamaga synanyşyk edilmegi onun golaýdaky az bitin sana çenli (şu ýagdaýda 4-e çenli) tegeleklenmegine getirýär we bu ahmiýet saklanýar. Umuman alnanda, bu size gerek netijedir, oboý zolaklarynyň kesindilerini äpisingäniň aşagyna ýa-da gapynyň ýokarysyna ýelmap boljak hem bolsa, olary rulona zerurlyk hasaplananda göz önünde tutmasaň gowy bolar.

Bir tipli ahmiýeti beýleki tipe öwrülmegi getirmek (çasting) diýlip atlandyrylýar. Bu anyk ýagdaý aýyk däl getirmegin (implicit çast) mysaly bolup durýar, sebäbi kodda getirmegin zerurdygy we kompilyatoryň ony özbaşdak etmelidigi aýyk görkezilen däldir. Kompilyasiýa wagtynda alýan iki duýdurysyňyz aýyk däl getirmä degişlidir, ol bir tipiň beýleki, takyklygy pes tipe öwrülmegi sebäpli maglumatlaryň bir böleginiň yitirilmegine getirip biler.

Aýyk däl getirmäni örän seresap ulanmalydyr. Kompilyatorlar sular ýaly getirmelerin ýerine ýetirilýandigi barada kähalatda duýduruş bermeyärler we eger siz uýtgeýan ululygyň bir tipine ýol berilýan ahmiýetleriň pes diapazonly tipi bolan iýtgeýan ululyga berýan bolsaňyz, onda şu hili ýagdaýlarda hemişe maglumaty yitirmek töwekgelçiligi bolýar. Eger siziň programmaňyzda bilgeşleýin goşulmadyk aýyk däl getirmeler bar bolsa, olar yüze çykarmagy kyn bolan yalnyşlara sebäp bolup biler.

Yöne sular ýaly ahmiýet bermegiň gutulgysyzdygy sebäpli, kompilyatora bu ýerde tötänleýin zadyň ýokdugyny we ol siziň maksat edinen zadyňyzdygyny görkezmek üçin, getirmegi aýyk görkezin. Bu operatoryň sag böleginde `int` tipine ähmiýeti aýyk getirmek bilen amala asyrylýar, ýagny operator sular ýaly bolýar. `strips per^roll = static_çats <int> (rollLength / heght); //Rulondaky zolaklaryň sanyny almak`

Sag tarapyndaky aňlatmanyň daşynda ýaýlar bilen `static_çats <int>` böleginiň goşulmagy siziň aňlatmanyň ahmiýetini `int` tipine özgertmek isleýändigizi kompilyatora aýyk habar beryär. Bu siziň oňki ýaly ahmiýetiň drob bölegini ýitirýändigizi aňladýan hem bolsa, kompilyator siziň nâme edýändigizi düşünyar we bu gezek duýduruş bermez. Siz şu bapda bu barada we aýyk getirmegin beýleki tipleri köp zat bilim bilersiniz.

Şu operatorda otagyň perimetriniň nähili hasaplanyp çykarylýandygyna üns beriň. `length` we `width` uýtgeýanleriň jemini 2-ä köpeltmek üçin jemiň aňlatmasy ýaý icine alynýar. Bu perimetriň dogry ahmiýetini almak üçin gosmagyň birinji ýerine ýetiriljekdigini, netijaniň bolsa 2.0-ä köpeldiljekdigini kepillendiryär. Ýaýyň kömegi bilen hasaplamagyň gerek bolan tertipde ýerine ýetiriljekdigini kepillendiryär, sebäbi ýaý icindaki aňlatmalar hemişe birinji ýerine ýetirilýar. Ýaý

icinde biri-biriniñ icinde goýlan bimaçe anlatmalar bolan ýagdaýynda bu anlatmalar yzygiderli - içerki yaylardan daşarky yaylara tarap tertipde hasaplanýar.

Otaga ýelmemek üçin zerur bolan oboý zolaklarynyň sanyny hasaplap çykaryan üçünji operator birinji operatora göreň effektiňiz ýaly effektden peýdalanýar. Netije az tarapyna golaýdaky bitin sana çenli tegeleklenýär, sebäbi ol strips regd bitin sanly uýtgeýan ululyga berilmelidir. Yöne bu hakykatdan size gerek zat däl. Köp tarapyna tegeklekse gowy bolardy, yöne siziň şeýle etmek üçin C++ dilindäki biliminiz ýeterlik däl. Indiki baby okap siz muňa gaýdyp gelip hem-de degişli düzedişleri girizip bilersiniz.

Soňky arifmetiki anlatma zolaklaryň sanyny (bitin san hökmünde) rulondaky zolaklaryň sanyna (ol hem bitin san hökmünde) bölmek bilen rulonlaryň zerur bolan sanyny hasaplaýar. Bir bitewiligi beýleki bitewilige bolýandigiňiz sebäpli, netije hem bitin san bolar we islendik galyndy inkâr ediler. nrolls uýtgeýan ululygyň uýtgeýan oturly tipi bolanda hem sonun ýaly bolar. Bölmek aňlatmasyndan alnan bitin ahmiýet uýtgeýan oturly ähmiýete öwrüler we nrolls uýtgeýan ululykda saklanar. Şeýle bolanda hem bu siziň diýýäniňiz däl, sonun üçin hem eger bu programmany peýdalanmak isleseňiz oňa düzediş girizmeli bolarsyňyz.

Netijäni görkezmek

Hasaplamalaryň netijesi şu operatoryň kömegi bilen görkezilýär.

```
cout << endi << "Siziň otagyňyza oboý ýelmap çykamak  
üçin, <<nrolls<< „rulon gerek bolar.“ << endl; return 0;
```

Bu kodun üç setirine ýaýradylan bir çykyş operatorydyr. Ilkibaşda ol täze setiriň simwolyny, sonra bolsa „Siziň otagyňyza oboý ýelmap çykamak üçin“ tekst setirini çykaryar. Sondan soň nrolls uýtgeýäniň ähmiýeti gelyär, sonra ýene bir tekst setiri - „rulon gerek bolar.“ - gelyär. Görşünüz ýaly, C++ dilinde çykyş operatoryny yazmak örän ýeňildir.

Programma şu operator bilen tamamlanýar

```
return 0;
```

Bu ýerde 0 - gaýtarylýan ähmiýetdir, ol şu ýagdaýda amal ulgamyna berilýär.

Gaýtarylýan ulgamlar barada siz 5-nji bapda giňisleyin bilersiniz. Galyndylary **hasaplamak**

Siz soňky mysalda bir bitin ahmiýetiň beýleki bitin ähmiýete bölünmegiň galyndyny inkâr etmek bilen bitin san beryändigini gördünüz. Sonun üçin hem eger 11-i 4-e bölsen, onda 2 alnar. Bölünenden soň galyndylaryň degerli gyzyklanma döredip biljekdigi sebäpli, mysal üçin, çagalaryň arasynda köke paýlananda, S++ dili munuň üçin yörite operatory - %- göz önünde tutýar. Başga sözler bilen aýdylanda kökani bölmek meselesi şu kody yazmak bilen çözüp bolar.

```
int residue = 0, çooies = 19, çildren  
= 5; residue = çookies % çildren;
```

residue uýtgeýan ululygy 4 ähmiýeti alar - bu 19-y 5-ä bölmekden galan sandyr. Her çaganyn naçe köke aljakdygyny hasaplap çykarmak üçin siz dine bölmekden peýdalanyň.

```
each = cookies / children;
```

Uýtgeýan ululygyň ahmiýetiniň uýtgemegi

Uýtgeýan ululygyň şu wagtky ahmiýetini uýtgetmek zerurlygynyň yüze çykýan wagty seýrek däl, şunda ony artdyrmak ýa-da iki esse köpeltmek gerek bolup biler, count uýtgeýan ululygyň ahmiýetini şu operatoryň kömegi bilen artdyryp bolar.

```
count = count + 5;
```

Bu ýerde count uýtgeýan ululygyň şu wagtky ähmiýetine 5 goşulyar we alnan netije yzyna count uýtgeýan ululyga ýerlesdirilýar, sonun üçin hem, eger ilki başda onuň ähmiýeti 10-a deň bolan bolsa, sondan soň ol 15-ä den bolar.

C++ dilinde şol ähmiýeti yazmak üçin alternatiw gysgaldylan sintaksis göz önünde tutulandyr.

```
count += 5;
```

Bu „count uýtgeýan ululygyň ahmiýetini almaly, oňa 5 goşmaly we alnan netijäni count uýtgeýan ululykda saklamaly" diýmegi aňladýar. Şunun ýaly notasiýada beýleki operatorlary hem peýdalanylýar. Mysal üçin.

```
count *= 5;
```

Bu count uýtgeýan ululygyň şu wagtky ahmiýetini 5-e köpeldýär we netijäni count uýtgeýan ululykda saklaýar. Umumy ýagdaýda operatorlary şu görnüşde ýazyp bolar.

```
çep op= sag;
```

çep bölegi ähmiýet bermegiň çep Döleginde ýerleşdirip boljak islendik dogry aňlatmany görkezýär we adaty (yöne hökmany däl) bu uýtgeýan ululygyň adydyr, sag bölegi - bu operatoryň sag böleginde ýerleşdirip boljak islendik dogry aňlatmadyr. op bölegi - bu aşakdaky operatorlaryň islendigi bolup biler.

Siz birinji baş operatora ön hem gabat gelensiniz, galanlary (süýşürme operatory we logiki operatorlar) bilen şu bapda tanyşarsynyz. Bu operatoryň umumy görnüşi şuna ekwiwalentdir.

```
çep = çep op (sag);
```

sag böleginiň daşyndaky ýaýlar bu aňlatmanyň birinji bolup hasaplanýandygyny, soňra op operatorynyň sag operandy bolýandygyny görkezýär. Bu bolsa siziň şu operatorlary ýazyp biljekdigiňizi aňladýar.

```
a /= b + c;
```

Bu aşakda getirilýan operatoryň berjek netijesini berer

```
a = p / (b + c);
```

Seýlelikde a uýtgeýan ululygyň ahmiýeti b we c uýtgeýan ululyklaryň ahmiýetleriniň jemine bölüner, netije bolsa a uýtgeýan ululyga berler.

4.7. Inkrementa we dekrementa operatorlary

Biz indi adaty bolmadyk, *inkrementa we dekrementa* diýlip atlandyrylýan arifmetiki operatorlar bilen tanyşdyraýys. C++ diliniň duýpli peýdalanylysyna göre, siz olary örän peýdaly hasap etjekdigiňize ynanýarsyň. Bu unar operatorlary bolup, olar uýtgeýan ululykda saklanýan bitin san tipli ahmiýeti artdyrmak ýa-da

peseltmek üçin hyzmat edýarlar. Mysal üçin, count uýtgeýan ululygyň int tipi bar diýip hasap etsek, onda indiki üç operator şol bir netijäni berer.

```
count = count + 1;  
count += 1;  
++ count;
```

Olaryň her biri kount uýtgeýan ululygyň ahmiýetini 1dereje artdyrýar. Inkrementa operatoryny peýdalanýan soňky görnüş hiç bir şübhesiz iň gysga görnüşdir.

inkrementa operatory dine bir özüniň ol babatynda ulanylan uýtgeýan ululygyň ahmiýetini uýtgetman, eýsem özi ahmiýeti gaýtaryp berýar. Şeýlelikde, uýtgeýan ululygyň ahmiýetini artdyrmak üçin inkrementa operatoryny peýdalanmak bilen, ony has çylşyrymly aňlatmanyň bir bölegine öwürup bolar. Eger ++ operatoryny peýdalanmak bilen, ++count aňlatmasynda bolşy ýaly, beýleki aňlatmanyň düzüminde artdyrsak, onda operatoryň hereketleri uýtgeýan ululygyň ahmiýetiniň ilki başda artmagyndan, soňra bolsa bu täze ahmiýetiň täze aňlatmasynda peýdalanylmagyndan ybarat bolar. Mysal üçin, count uýtgeýan ululygy 5 ahmiýeti öz icine alýar, siz hem int tipli total uýtgeýan ululygy kesgitlapsiňiz diýeliň. Şeýle hem siz şu operatory ýazypsyňyz diýip hasap edeliň.

```
total = ++count + 6;
```

Bu count uýtgeýan ululygyň ahmiýetini 6-a çenli artdyrar we oňa 6 goşular, sonuň üçin total uýtgeýan ululyk 12 ahmiýeti alar.

Şu wagta çenli biz ++inkrementa operatoryny ol haýsy uýtgeýan ululyk babatynda ulanylan bolsa, şol uýtgeýan ululygyň önünde ýerlesdirdik. Bu inkrementanyň prefiks görnüşi diýlip atlandyrylýar. Emma bu operatoryň postfiks görnüşi hem bardyr, şeýle bolanda ol haýsy uýtgeýan ululyk babatynda ulanylan bolsa, şol uýtgeýan ululygyň adyndan soň ýerlesýar. Netijede, biraz başgaça effekt alynýar. Ol babatynda operator ulanylan uýtgeýan ululygyň ahmiýeti dine onuň ahmiýeti göwrümli kontekstde peýdalanylandan soň artrýar. Mysal üçin, count uýtgeýan ululyga 5 ahmiýeti gaýtaryp beriň we ozalky operatory şu görnüşde gaýtadan ýazyň

```
total = count++ + 6;
```

Netijede total 11 ahmiýeti alar, sebäbi, count uýtgeýan ululygyň başlangyç ahmiýeti inkrementa operatorynyň count uýtgeýan ululygyň ahmiýetini 1 derejede artdyrmazyndan ön hasaplamalar üçin peýdalanylýar. Bu operator aşakdaky iki operatora ekwiwalentdir.

```
total = count +  
6; ++count;
```

Yokarda getirilen postfiks görnüşiň mysalynda bolşy ýaly, isimwollarynda tygşytlylyk aýdyňlygy peseldýar. Düzgün bolşy ýaly, inkrementa operatoryny meniň şu ýerde ýazysym ýaly ýazmaly däl. Yazmak üçin şu görnüş peýdalanylsa gowy bolar.

```
total = 6 + count++;
```

Yöne haçanda sizde $a++ + b$ ýa-da $a+++b$ aňlatmasy bar bar bolsa, onda onuň nâme aňladýandygy we kompilýatoryň nâme etjekdigi aýdyň däl. Hakykatdan bu şol bir zatdyr, yöne ikinji ýagdaýda biz $a + ++b$ göz önünde tutup bilerdik, bu bolsa eýýam şol bir zat däl.

Meniň inkrementa operatory bilen bagly gürrün beren kadalarym - - dekrementa operatory babatynda hem ulanarlyklydyr. Mysal üçin, count uýtgeýan ululygyň ilkibaşky ahmiýeti 5-e deň diýeliň.

```
total = - count + 6;
```

Sonuň netijesinde uýtgeýan ululyk total 10 ahmiýeti alar, indiki anlatma bolsa oňa 1 1 ähmiýet berer.

```
total = 6 + count - - ;
```

Düzgün bolşy ýaly, iki operator hem bitin sanly uýtgeýan ululyklar babatynda ulanylýar, adatyça siklerde ulanylýar, bu 3-nji bapda görkeziler. Beýleki baplardan siz C++ dilindäki maglumatlaryň beýleki tipleri babatynda, mysal üçin, ýadyň salgylaryny saklayan uýtgeýan ululyk babatynda ulanylyp bilner.

Amaly ders "Otur" operatory

"Otur" (.) operatory adatyça aňlatmanyň biriniň bolup biljek ýerinde birnäçe aňlatmalary ulanmaga mümkinçilik beryär. Bu ýagdaýy amaly mysalda has gowy düşündirip bolar.

```
//Ex2_0_6.cpp
// Otur operatory bilen yumuş (gönükme)
// #include <iostream>
using namespace std;
int main ( )
{
    long num1 = 0, num3 = 0, num4 = 0; num4 = (num1 =
    10, num2 = 20, num3 = 30); cout << endl<<
    "Anlatmalar tapgyrynyň ähmiýeti" << "olaryň iň
    sagdakysynyň ähmiýeti bolup durýar:" << num4 ; cout
    << endl; return 0;
}
```

Ex2_0_6.cpp kodunyň fragmenti

Bu nähili isleyär?

Bu programmany kompilyasiýa edip, işe goýbersek, onda şu netijäni alarys.

Anlatmalar tapgyrynyň ähmiýeti olaryň iň sagdakysynyň ähmiýeti bolup durýar: 30

Bu düşündirişleri talap etmeýar. main() funksiýasynyň birinji operatory num1-den num4-e çenli dört uýtgeýan ululygy döredýär we funksiýanyň notasiýasynda olary nol ähmiýet bilen inisiallasdyrýar. Num 4 uýtgeýan ululyk berlen üç ahmiýetiň iň soňkusynyň ahmiýetini alýar, onuň çep böleginiň alýan ähmiýeti bolsa berilyän ähmiýet bolar. Num4 uýtgeýan ululyga ähmiýet bermek

amalynda ýaýlar möhümdir. Bu mysaly ýaýlary peýdalanman ýerine ýetirjek boluň we näme bolýanyňa syn ediň. Yay icine alynmadyk, otur bilen bölünip aýrylan birinji anlatma sular ýaly bolar. num4 = num1 = 10L;
ýagny num4 uýtgeýan ululyga 10L ähmiýet berler.

Elbetde, „otur“ operatory bilen bölünip aýrylan anlatmalar berlen bolmaly däldir. Siz edil sonuň ýaly, şu operatorlary hem ýazyp bilersiniz.

```
long num1 (1L), num2 (10L), num3 (100L), num4 (0L);
```

Bu ähmiýet beriş operatorynyň netijesi num1, num2 we num3 uýtgeýan ululyklaryň ahmiýetlerini 1 dereje artdyrmak, şeýle hem num4 uýtgeýan ululyga soňky anlatmanyň ahmiýetini bermek bolar, şu ýagdaýda soňky ähmiýet 101 bolar. Bu my sal „otur“ operatorynyň hereketini görkezýär, yöne ol gowy kódy ýazmagyň mysaly bolup durmaýar.

Hasaplamalaryň yzygiderliligi

Şu wagta çenli men anlatmada hasaplamalaryň yzygiderliligine nähili täsir etmelidigi barada gürrün etmedim. Bu esasan, sizin mekdepde okan wagtyňyzda tanyş bolan esasy arifmetiki amallaryňyza laýyk gelýar, yöne C++ dilinde başga-da köp amallar bardyr. Olar bilen näme bolup geçjekdigine düşünmek maksady bilen bu yzygiderliligi kesgitlemek üçin C++ dilinde peýdalanylýan we operatorlaryň prioriteti diýlip atlandyrylýan mehanizme serede!iň.

Operatorlary ýerine ýetirmegin tertibi

Ýerine ýetirmegin tertibi operatorlary prioritetler boyunça bölyär. islendik anlatmada ýokary prioriteti bolan operatorlar hemişe birinji ýerine ýetirilýar, olaryň yzyndan prioritet derejesi ondan pes bolan operatorlar ýerine ýetirilýar. C++ dilinde operatorlary ýerine ýetirmegin tertibi 4.4 tablisada berilyär.

C++ diliniň operatorlaryny ýerine ýetirmegin tertibi

4.4 tablica

<i>Operator</i>	<i>Assosiativlik</i>
::	Çep
0 1J->•	Çep
! ~ i- (un ar) - (unar) ++ -- 7& (unar) * (un ar) typecast (tip in getirilişi) static çast const çast dynamic çast reinterpret çast sizeof new çielete typeid decltype	Sag
. * (unar) ->*	Çep
* / %	Çep
+	Çep
<< >>	Çep
< <= > >=	Çep
1 =	Çep
&	Çep
A	Çep

!	Çep
&&	Çep
II	Çep
?: (şertli operator)	Sag
<<= >>=	Sag
<i>t</i>	Çep

Bu ýerde köp operatorlar getirilýär, siz haziriikçe olar bilen tanyş dălsiniz, yöne şu kitabyň ahyryna çenli olaryň hemmesini tanarsyňyz. Olary baplar boyunça bölüşdirmegin ýerine biz C++ dilindăki ähli operatorlary ýerine ýetirmegin tertibiniň bir tablisasyna jemledik, bir operatoryň prioritetiniň beýlekiniň prioriteti bilen năhili gatnasygynyň bardygy babatynda şübhe dörän wagtynda siz hemişe oňa ýuzlenip bilersiniz.

Iň ýokary prioriteti bolan operatorlar tablisanyň ýokarysynda ýerleşýarler. Tablisanyň bir öyjünde görkezilen ähli operatorlaryň birmeňzes prioriteti bardyr. Eger anlatmada ýaý ýok bolsa, onda deň prioriteti bolan operatorlar olaryň assosiatiwligi bilen kesgitlenýan yzygiderlilikde ýerleşýarler. Başga sözler bilen aýdylanda, assosiatiwlik „çep“ bolsa, onda aňlatmadaky iň çepdaky operator birinji ýerine ýetirilýar, soňra ähli aňlatmanyň operatorlary çepden saga tertipde ýerine ýetirilýar. Bu $a + b + c + d$ ýaly aňlatimalaryň ýazylysy ýaly, ýagny $((a + b) = c) = d$ ýaly ýerine ýetirilýändigini aň lady ar, sebäbi $+$ binar operatorynyň çep assosiatiwligi bardyr.

Operatoryň iki görnüşinin, ýagny unar (bir operand bilen işleyar) we binar (iki operandl) bar mahalynda unar görnüşiniň has ýokary prioritetiniň bardygyna wç sonuň üçin hem birinji ýerine ýetirilýändigine üns berin.

Bellik. Yaýlaryň kömegi bilen anlatmada operatorlary ýerine ýetirmegin tertibini hemişe üytgedip bolar. C++ dilinde örän köp operatorlaryň bardygy sebäpli, kăhalatlarda çylşyrymly aňlatmany hasaplamagyň tertibiniň năhili boljakdygyna düşünmek örän kyn bolýar. Sonuň üçin hem yaýlaryň ulanylmagy ynam döretmäge mümkinçilik beryär. Yaýlary ulanmakdan alynýan goşmaça peýda kôdy okamagyň yeňillesýandigi bolup durýar.

Tipleri özgertmek we getirmek

C++ dilindăki hasaplamalar dine bir tipli ahmiýetleriň arasynda ýerine ýetirilip bilner. Dürli tiplerin uýtgeýan ululyklaryny we konstantal ary ny öz icine alýan anlatma ýazylýan wagtynda ýerine ýetirilýan her bir amal üçin kompilyator bir operandyň tipini beýleki operandyň tipine özgertmelidir. Tipleri özgertmek işi getirmek diýlip atlandyrylýar. Mysal üçin, double tipiniň ähmiýeti bitin sanly tipin ähmiýetine goşmak isleseňiz, onda bitin sanly ähmiýet ilkibaşda double tipiniň ähmiýetine öwrülýär, sondan soň goşmak ýerine ýetirilýar.

Elbetde, başlangyç ähmiýeti özünde saklayan uýtgeýan ululygyň özi uýtgemeyar. Kompilyator üytgan ähmiýeti ýadyň wagtlaýyn öyjünde saklaýar, hasaplamalar tamamlanandan soň ol boşayar.

islendik amalda özgerdilmäge degişli operandy saýlap almak berk kadalar bilen kesgitlenýar. işlenip geçilmeli islendik anlatma iki operandly amallaryň

tapgyryna bölünýar. Mysal üçin, $2 * 3 - 4 + 5$ aňlatmasy şu tapgyrlardan durýar: $2 * 3$ netijede 6 beryär, soňra $6 - 4$, bu netijede 2 beryär we iň sonunda $2 + 5$, onuň netijesinde -7 alynýar.

Şeýlelikde, operandlary getirmegin kadalary dine operandlaryň jübütligi babatynda çözgüt kabul etmek üçin zerurdyr. Sonun üçin hem dürli tipli operandlaryň islendik jübüti üçin aşakda getirilýan prioritetler barlanýar. Sanawda tipi aşakda ýerleşen operand sanawda tipi ýokarda görkezilen operanda getirilyär.

1. long double
2. double
3. float
4. unsigned long long
4. long long
7. unsigned long
8. long
9. unsigned int
10. int

Seýlelikde, eger sizde long long we unsigned int tipli operandlar bar bolsa, onda olaryň soňkusy long long tipine öwrüler. char, signed char, unsigned char, short ýa-da unsigned short tipli islendik operand amalyň oňýanynda iň bolmanda int tipine öwrulyär.

5. OPERATORLAR BARADA DÜŞÜNJE, GÖRKEZIJILER WE SALGYLAR

5.1. Operatorlar we bloklar

Eger islendik anlatmadan soň nokatly otur bar bolsa (mysal üçin, $x=0$, $i++$ ýa-da `printf (...)`) bar bolsa, onda şunun ýaly anlatma C dilinde operator diýlip atlandyrylýar.

Seýlelikde:

```
x = 0; - operator;  
LJ i++; -operator;  
printf (...) ; - operator.
```

Şekilli ýaýlar operatorlary bloklara birleşdirmek üçin peýdalanylýar. Şunun ýaly blok sintaktika taýdan bir operatora ekwiwalentdir (deňdir), yöne nokatly otur blokdan soň goýulmaýar (kompilýator blogyň gutarýan ýerini ony ýapýan ýaý boýunça kesgitleýar.

Şu kesgitlemeden ugur almak bilen, operatorlaryň tabşyryk biçuwini şunun ýaly ýazyp bolar:

For (aňlatma)	while (anlatma)	if (anlatma)
blok	blok	blok

Eger blokda dine operator bar bolsa, onda şekilli ýaýlary görkezmän bolar.

Blokda uýtgeýan ululyklary ygılan edip bolar, yöne olaryň lokal (ýerli), ýagny, blogyň daşynda tanalmaýan ululyklar bolup durýandygyny unutmaly däl.

Operatorlar programmany ýerine yetirmek prosesini dolandyryrlar. C++ dilinde operatorlaryň toplumy düzümläýin programmirlämegiň ähli dolandyryş kontruksiýalaryny öz içine alýar.

Düzüm operatory şekilli ýaýlar bilen çäklendirilýär. Beýleki operatorlaryň ählisi nokatly otur bilen gutarýar.

***Boş operator -;**

Boş operator - bu dine nokatly oturdan durýan operatordyr. Ol Drogrammanyň sintaksis boýunça operatoryň talap edilýan islendik yerinde peyda r>olup biler. Boş operatoryň yerine ýetirilmegi programmanyň ýagdaýyny tgetmeyär.

***Düzüm operatory - {...}**

Düzüm operatorynyň hereketi özünde bar bolan operatorlary yzygiderlii лте ýetirmekden ybaratdyr, haýsydyr bir operatoryň dolandyryşy >rogrammanyň beýleki ýerine açyk berýan halatlary muňa girmeyär.

Kadadan çykmalar işlap geçyan operator

- **try** { <ope za torlar> }
- **catch** {<kadadan çykmalary ygılan etmek >} { < operatorlar > }
- **catch** [<kadadan çykmalary ygılan etmek >] { < operatorlar > }
- ...

- **catch** [*<kadadan çykmalary yglan etmek >*] { *<operatorlar>* }

Şertli operator

- **if** (*<aňlatma>*) *<operator1>* [**else** *<operator 2>*]

Gaýtadan utgaşdyryjy operatory

- **switch** (*<aňlatma>*)
- **↓ case** *<hemişelik anlatma 1>*: *<operatorlar1>*
- **case** *<hemişelik anlatma 2>*: *<operatorlar 2>*
- **case** *<hemişelik anlatma N>*: *<operatorlar N>*
- [**default:** *<operatorlar >*]

Gaýtadan utgaşdyryjy operator programmany yerime yetirmegin birnäçe alternativ/ ýollaryndan birini saýlap almak üçin niyetlenendir. Gaýtadan utgaşdyryjy operatoryň hasaplamak aňlatmany okamakdan başlanyar, sondan soň dolandyryş konstant anlatma bilen bellenen, aňlatmanyň okalan ahmiýetine deň bolan operatora berilýar. Gaýtadan utgaşdyryjy operatorlardan çykmak break operatory bilen tarapyndan amala asyrylýar. Eger aňlatmanyň ahmiýeti konstant ahmiýetleriň hiç birine deň bolmasa, onda dolandyryş default esasy söz bilen bellenen operatora berilýar (eger ol bar bolsa).

Önünden şertli siki operatory

- **while** (*<aňlatma>*) *<operator>*

Sonundan şertli siki operatory

do *<operator>* **while** *<aňlatma>*;

C++ dilinde bu operator sonundan şertli sikliň nusgawy durmuşa geçirilişinden aňlatmanyň hakyky bolan wagtynda siklden çykmak däl-de, sikliň işinin dowam etmeginin bolup geçyandigi bilen tapawutlanýar.

- **for** ([*<başlangyç anlatma >*];
- [*<şertli anlatma >*] ;
- [*<artdyrma aňlatmasy >*])
- *<operator>*

for operatorynyň bedeni şertli aňlatmanyň ýalan anlatma bolýan (nola deňleşýän) wagtyna çenli ýerine ýetirilýar. Başlangyç anlatma we artdyrma anlatmak adatça sikliň parametrlerini we beýleki ahmiýetleri inisiallaşdyrmak we modifisirlemek üçin peýdalanylýar. Başlangyç ähmiýet bir gezek, şertli aňlatmanyň birinji gezek barlanýan wagtyna çenli hasaplanýar, artdyrma aňlatmasy bolsa operator her gezek ýerine ýetirilenden soň hasaplanýar. Sikliň sözbaşylarynyň üç aňlatmasynyň her biri, hatda üçüsi hem goýberilip (görkezilmän) bilner (dine nokatly otur goýmany ýatdan çykarmaň). Eger şertli anlatma goýberilen bolsa, onda ol hakyky hasap edilýar we siki tükeniksiz bolýar.

Adimleýin sikliň operatory C++ dilinde çäýe we amatly kolnstruksiýa bolup durýar, sonun üçin hem önünden şertli while siki operatory C++ dilinde örän

seýrek peýdalanylýar, çünki köp ýagdaýlar for operatoryndan peýdalanmak amatly bolup durýar.

Break-bölüji operatory

Bölüji operator **while**, **do**, for we **switch** operatorlarynyň ýerine ýetirilmeginiň arasy kesýar. Ol dine şu operatorlaryň bedeninde bolup biler. Dolandyryş arasy kesilen operatorlardan soňky programma operatoryna berilýar. Eger bölüji operator **while**, **do**, for, **switch** operatorlarynyň icinde ýazylan bolsa, onda ol dine özüni icine alýan operatory tamamlýar.

continue dowam etdiriş operatory

Dowam etdiriş operatory dolandyryşy **while**, do, for sikl operatorlaryndaky nobatda iterasiýa berýar. Ol dine şu operatorlaryň bedeninde bolup biler, do we while operatorlaryndaky nobatdaky iterasiýa şertli aňlatmany hasaplamakdan baslanýar. for operatorynda nobatdaky iterasiýa artdyрма aňlatmasyny hasaplamakdan baslanýar, soňra bolsa şertli aňlatmany hasaplamak bolup geçýar.

Gaýtarys operatory - return<aňlatma>;

Gaýtarys operatory özüniň bar bolan funksiýasyny ýerine yetirmegi tamamlýar we dolandyryşy çagyryň funksiýa berýar. Dolandyryş çagyryş operatorynyň gönüden-göni yz ýanyndaky çagyryň funksiýanyň nokadyna berilýar.

Aňlatmanyň ahmiýeti, eger ol berlen bolsa, hasaplanýar, gaýtarýan operatory özünde saklayan funksiýasy üçin yglan edilen tipe getirilýar we çagyryň funksiýa gaýtarylýar. Eger aňlatma görkezilmedik bolsa, onda funksiýa tarapyndan gaýtarylýan ahmiýet kesgitli däl.

Formai nukdaýnazardan **break**, continue we return operatorlary gurluşy programmirlenmegiň operatorlary bolup durýarlar. Emma olaryň programına düşünmegi ýeňillesdirýän we icinde gurlan uly gurluşdan gaça durmaga mümkinçilik berýän ýagdaýlarynda olaryň çakli möçberlerde peýdalanylmagy esasydyr. Mysal üçin, biz giriş maglumatlaryny anomaliýalar babatynda barlaýarys.

Eger bu operatorlary peýdalanmasaň, onda ähli islöp geçmeleri şertli blogyň icinde goýup bolar, bu bolsa programmanyň okalyşyny ýaramazlasdyrýar. Munuň ýerine başlangyç maglumatlar ýalňys bolanda funksiýadan çykmagy guraýan uly bolmadyk şertli blogy ýazyp bolar.

Sizin programmada yglan edýän her bir uýtgeýän ululygyňyzyň **salgysy** - ýatda özüniň ýerlesýän öýjüginin belgisi bardyr. Salgy uýtgeýän ululygyň aýrylmaz hasiýetnamasydyr. Bu ýazgyny saklayan başga bir uýtgeýän ululygy hem yglan edip bolar, ol **görkeziji** diýlip atlandyrylar. Görkezijiler biziň üytgetmek isleýän parametrlerimiz funksiýa bedende, massiwler bilen işlenende, dinamiki ýat bilen işlenende we beýleki birnäçe halatlarda ulanylýar.

Görkezijini yglan etmegiň şunun ýaly sintaksisi bardyr:

<lipi> *(<identifikator> [= <inisiallaşdyryjy>]);

Görkeziji binýatlyk, geçirilýän tip in	// görkeziji int
char *ppç;	// görkezijä görkeziji char

<code>int* p, s;</code>	<i>// nädogry yglan etme , s... görkeziji bolmavar!</i>
<code>i n t * p, s ;</code>	<i>// s ... görkeziji bolmavar</i>
<code>int * p, * s ;</code>	<i>// iki görkeziji</i>
<code>char *names[] = { "John", "Anna" };</code>	<i>//görkezij ilerin massiw i</i>

Soňky yglan etmede tipi emele getirmek üçin iki operator peýdalanylýar: * we | |, olaryň biri adyň önünde, beýleki ondan soň durýar. Olar prefiks ýa-da suffiks bolan bolsa, onda yglan etme operatori ary ndan peýdalanmak ep-esli ýeňilleserdi. Emma * we [] we () olaryň manysyny anlatmalarda görkezmek üçin işlenip taýýarlanypdy. Seýlelikde, * prefiks, [] we () bolsa suffiks bolup durýar. Suffiks operatorlary prefikse garanyňda atlara "has berk baglydyr". Diýmek, *names[] haýsydyr bir obýektleri görkezyän görkezijiler massiwidir, "funksiýalary görkeziji" ýaly tipleri kesgitlemek üçin bolsa ýaýlar peýdalanylmalydyr.

Görkezilen bilen işlenmegine degişli bolmadyk iki amal bardyr. Bu amallar sulardan ybaratdyr:

*salgyny almak amaly (salgylaşdyrma) &,*salgy boyunça ähmiyeti almmak amaly (gytaklaýyn salgylaşdyrma ýa-da adyny aýyrmak) *.

`i n t a,* p ;`

`P = &a`

`*P= 0;`

// a uýtgeýan ululyga p uýtgeýan ululygyň salgysy berilýar

// p uýtgeýan ululygyň salgysynda durýan baha 0 berilýar

void görkeziji

void tipini görkezij ilerin yörite ulanylyşy bardyr. void tipini görkeziji islendik tipin ähmiyetini görkezip biler. Emma void görkezij isinin ýa-da görkezilyän obýektiň üstünde amallar geçirilende görkezij inin tipini void görkezij i sinden tapawutlanýan tipe açyk getirmek zerurdyr.

islendik tipdäki obýekti görkezij ini void* tipli uýtgeýan ululyga berip bolar, bir void* beýleki void* berip bolar, iki void* dendigi ýa-da deň dældigi babatda deneşdirip bolar we void* başga tipli görkezijä açyk öwürüp bolar. Gaýry amallar howply bolup biler, sebäbi kompilýator görkezijinin hakykatdan haýsy görnüşdäki obýekte salgylanýandygyny bilmeýar. Sonuň üçin hem amallar kompilýasiýa tapgyrynda ýalňyslyk baradaky habary çagyryýar. void * peýdalanmak üçin ony beýleki tipi görkezijä açyk öwürmek gerek. vo id f (int * p i)

`void *pv -`

`*pv;`

`pv+ t`

`i n t *pi2`

p i ;

```
// int*ululygy void* ululyga özgertmek II ýalňys void* aýyrmak bolanok II ýalňys
inkrement edip bolanok void* II int* ululygyň açyk özgerdilişi
static çast<int*>(pv) ;
    double *pdl = pv;                //ýalňys
    double *pd2 = pi;                //ýalňys
    double *pd3 =                    //howply
static çast<double*>(pv);
```

Düzgün bolşy ýaly, görkezijini onuň görkezyän obýektinden tapawutlanýan obýekt tipine öwürmek görkezijini peýdalanmak howply bolup biler.

void* esasy ulanylyşy görkezij ileri obýektleriň tipi barada çaklamalar etmäge rugsat berilmeyän funksiýalara bermek, edil sonun ýaly hem funksiýalardan „takyklanmadyk“ tipi obýektleri gaýtarmak bolup durýar. Şunun ýaly obýektiden peýdalanmak üçin görkezijiniň tipini açyk özgertmek zerurdyr.

void* görkezij ileri peýdalanýan funksiýalar, adatça ulgamyň iň pes derejelerinde, apparat serişdeleri bilen işlerin geçirilýan yerinde bolýarlar. Yokary derejelerde void* bolmagy şübhelidir we bu taslanylýan tapgyrda indikator tarapyndan goýberilen ýalňyslyk bolmagy ähtimaldyr.

5.2. Çörkezijiler we konstantalar

Görkezijiler bilen amallara iki obýekt - görkezijniň özi we onuň salgylanýan obýekti gatnaşýar. *const* esasy sözünin görkezijini ygylan etmegiň önünde yerleşdirilmegi görkezijini däl-de obýekti konstanta öwürýär. Görkezijniň özüni konstanta hökmünde ygylan etmek üçin dine * däl-de *const ygylan etme operatory peýdalanylýar. voіç f (char *p)

```
{ char s[] = "const";
const char *pl = s;                // konstanta görkezij i
pl[3] = 'r';                      // ýalňys - pl konstanta görkezyär
pl - p;                          // dogry
char * const p2 = s;              // konstanta görkezij i (inisializasiýa zerurdyr)
p2[3] = 'r';                     // dogry
pl - p;                          // ýalňys - p2 konstanta
    const char * const p3 = s;    //konstanta görkezijä konstanta
p3[3] = 'r';                     // ýalňys - p3 konstanta görkezyär
p3 = p;                          // ýalňys - p3 -konstanta
```

Uýtgeýan ululygyň salgysyny konstantany görkezijä berip bolar, sebäbi bu zyýansyz amaldyr. Yöne konstantanyň salgysyny erkin görkezijä berip bolmaz, sebäbi şeýle ýagdaýda konstanta obýektiniň ahmiýetini uýtgedip bolar.

Salgy arifmetikasy

Görkezij ilerin üstünde şu hareketleri geçirip bolar: ■

görkeziji bilen bitin sany goşmaly;

görkezij iden bitin sany aýyrmaly;

■ görkezijden görkezij ini aýyrmaly - netijede bitin san emele getirýär. Şundan gosulýan/aýrylýan ýa-da alynýan bitin san baýtyň sanyny däl-de, görkezijiniň görkezyän tipinin elementlerinin sanyny aňladýar, ýagny bu san tipiň möçberine köpeldilyär ýa-da bölünýar.

Görkezijleri biri-birinden aýyrmak dine iki görkezijinin hem şol bir massiwiň elementlerini görkezyän ýagdaýynda (dil munuň şeýledigini çalt barlamaga mümkinçilik bermeyän hem bolsa) bellendir. Bir görkezijini beýlekisinden aýyrmagyň netijesi massiwiň bu görkezij ilerin arasyndaky elementlerinin sany (bitin san) bolar. Bitin sany görkezijä goşmagyn we bitin sany görkezij iden aýyrmagyň netijesinde täze görkezij i alynýar. Eger şunun ýaly görnüşde alnan görkezij i massiwiň başlangyç görkezijinin görkezyän şol bir elementini (ýa-da ondan soňky elementi) görkezmeyän bolsa, onda görkezijini peýdalanmagyň netijesi kesgitlenen däl-dir.

Görkezijleri massiwleri islöp geçmek üçin peýdalanyp bolar.

```
int a [100], n, *end, *p;
```

```
end = a + n;
```

//n - a massiwiň elementlerinin sany.

Massiwiň ady onuň başlangyjynyn salgysy

bolup durýar

// şeýlelikde, end - massiwiň soňky

elementinden soň durýan elementin salgysy for

```
(p = a; p < end; p++)
```

```
printf("%4d", *p);
```

Görkezij iden bitin sany aýryp bolýandygyndan amalda otrisatel sanlary peýdalanmagyň mümkindigi gelip çykyar [].

```
int a [N] ;
```

```
int *endA = a + N - 1, i;
```

```
for (i = 0; i < N; i++)
```

```
printf("%4d", endA[-
```

```
i]);
```

Salgylanmalar

Salgylanma obýektiň alternativ/ ady bolup durýar. Salgylanma - bu sintaktika taýdan uýtgeýan ululyk ýaly görünyar, semantika boyunca salgy bolup durýar. Salgylanmany yglan etme inisializatory öz icine almalydyr, salgylanmanyň funksiýanyň parametri, funksiýa tarapyndan gaýtarylýan ahmiýet ýa-da derejaniň agzasy bolup durmayan halatlary muňa girmeýar. Soňra amallaryň ählisi salgylanmanyň üstünde däl-de, onuň görkezyän obýektiniň üstünde geçirilyar. int a - 10;

```
int &r - a;
```

//salgylanmany yglan etmek

```
r++"
```

// a uýtgeýan ululyk 11 deň bolýar

```

void f(double &a) { a
+= 3.14; } double d =
0;
int v[20];
int& f(int i) { return v[i]; }
f(3) = 7;

```

//v[3] massiwıń elementi 7 baha berilyär

Birbada seredenińde salgylanma gorkezijiniń gowy oruntutary bolup durýar, yöne salgylanmanyń sintaksisi bilen semantikasynyń gabat gelmeyändigigi sebäpli programına düşünmegi kynlaşdyryar. Emma salgylanmalar uly möçberi bolan funksiýanyń parametrini ähmiyeti boýunça bermezlik ücin (nusgasyny almazlyk ücin) peýdaly bolup bilerler. Şu ýagdaýda, eger biz bu parametri funksiýanyń icinde üytgetmek islemeyän bolsak, onda const modifikatory bilen salgylanmany yglan edip bolar. Şu ýagdaýda biz parametriń uýtgemejekdigigi, uly obýektiń yerine onuń salgysynyń beriljekdigigi babatynda kepillik alyp bileris, peýdalanyjy ücin bolsa, bu parametrleri ähmiyeti boýunça bermek ýaly bolup görner.

```

çlass x { ... };
int f(const X& x) { ... }

```

//uly klasyń teswiri

6. FUNKSIÝALAR WE GÖRKEZİJİLER BİLEN İŞLEMEK. RUGSAT SPESİFİKATORY.

6.1. Funksiýalar, ony döretmek we çagyrmak

Funksiýalar - bu koduň programmanyň islendik yerinden birnäçe gezek çagyryp bolýan fragmentidir. Geçen baplarda biz standart kitaphananyň düzümine giryän, icinde gurlan funksiýalardan birnäçe gezek peýdalanyppydık. Mysal üçin, `strlen ()` funksiýasynyň kömegi bilen Ç-setirde simwollaryň sanyny aldyk. Bu bapda biz programma kodunyň artykmaçlygyny azaltmaga we onuň struktirlenişini ýokarlandyrmaga mümkinçilik berýan peýdalanyjylar funksiýalarynyň döredilişine serederis.

Funksiýany döretmek we ony çagyrmak

Funksiýanyň beýany iki bapdan durýar: yglan etmek we kesgitleme. Funksiýany yglan etmek (ol funksiýanyň prototipi diýlip hem atlandyrylýar) tip baradaky maglumatlary öz icine alýar. Bu maglumaty peýdalanmak bilen kompilyator parametrleriň tipi bilen sanynyň laýyk gelmezligini tapyp biler. Funksiýanyň prototipiniň biçiwi (formaty)

```
<netijaniň tipi> <funksiýanyň ady>([<tipi> [<parametriň ady>]  
[ , . . . , <tipi> [<parametriň ady n>] ] ) ;
```

"Netijaniň tipi" parametre ahmiýetiň tipini berýar, funksiýa bu ahmiýeti return operatorynyň kömegi bilen gaýtaryar. Eger funksiýa hiç hili ahmiýeti gaýtarmaýan bolsa, onda tipiň ýerine void esasy sözi görkezilyär. Funksiýanyň ady vol berilýan identifikator bolmalydyr, oňa uýtgeýan ululyklaryň adyna bildirilýan talaplar ýaly talaplar bildirilýar. Funksiýanyň adyndan soň tegelek ýaýlaryň icinde parametrleriň tipi we ady olaryň arasynda otur goýmak bilen görkezilyär. Funksiýanyň prototipinde parametrleriň adyny asla bermän hem bolar, çünki kompilyatory dine maglumatlaryň tipi we parametrleriň sany gyzyklandyrýar. Eger funksiýa parametrleri kabul etmeýan bolsa, onda dine tegelek ýaýlar görkezilyär ýa-da tegelek ýaýlaryň icinde void esasy sözi berilýar (Ç dilinde void esasy sözi hökmany bolup durýar). Funksiýa yglan edilenden soň nokatly otur durmalydyr.

Funksiýalary yglan etmegiň mysaly:

```
int sum(int x, int y) ; // ýada int sum(int, int) ;  
void print(const char *str) ; // ýada void print(const  
char*) ;  
void print_ok() ; // ýada void print_ok(void) ;
```

Funksiýanyň kesgitlemesi parametrleriň tipiniň we adynyň beýanyny, şeýle hem onuň durmuşa geçirilişini öz icine alýar. Funksiýanyň kesgitlemesiniň şunun ýaly biçuwi(formaty) bardyr:

```
<Netijaniň tipi> <funksiýanyň ady> - ([ <tipi> <  
parametriň ady 1>  
[ , . . . , <tipi> <parametriň ady  
N>] ] )
```

```
<fur.ks.i ýanyň bedeni>
return[<gaýLarýan
bahasy>];
```

Prototipden tapawutlylykda funksiýanyň kesgitlemesinde tipden soň hökmany suratda parametriň ady durmalydyr, ol lokal uýtgeýan ululyk bolup durýar. Bu uýtgeýan ululyk funksiýa çagyrylanda döredilýär, funksiýadan çykarylada bolsa ol aýrylýar. Seýleklikde, lokal uýtgeýan ululyk dine funksiýanyň içinde görünyar. Eger lokal uýtgeýan ululygyň ady global uýtgeýan ululygyň ady bilen gabat gelýan bolsa, onda ähli amallar lokal uýtgeýan ululyk bilen geçiriler, global uýtgeýan ululygyň ahmiýeti bolsa uýtgemez. Şu ýagdaýda global uýtgeýan ululyga yüzlenmek üçin adyň önünden :: operatory görkezmek zerurdyr. Mysal: `int sum(int x, int y) { int z = x + y; // x lokal uýtgeýan ululyga yüzlenmek z = ::x + y; // y global uýtgeýan ululyga yüzlenmek return z; }`

Parametrler beýan edilenden soň şekilli ýaýlaryň içinde instruksiýalar ýerlesdirilýar, olar her gezek funksiýa çagyrylanda ýerine ýetirilmelidir. Şekilli ýaýlar islendik ýagdaýda, hatda funksiýanyň bedeni dine bir instruksiýadan durýan hem bolsa, görkezilýär. Yapýan şekilli ýaýdan soň nokatly otur görkezilmeyär.

Funksiýadan ahmiýeti gaýtarmaga return operatory mümkinçilik berýar. Bu operator ýerine ýetirilenden soň funksiýany ýerine yetirmek saklanýar we dolandyryş yzyna, funksiýany çagyryş nokadyna berilýar. Bu bolsa return operatoryndan soň funksiýanyň hiç wagt ýerine ýetirilmejekdigini aňladýar. return operatory peýdalanylanda anyk däl ýagdaýlar bolmaly däldir. Mysal üçin, şu ýagdaýda gaýtarylýan ahmiýet şu serte baglydyr:

```
int sum(int x, int y) {
    if (x > 0) {
        return x + y;
    }
```

Eger x uýtgeýan ululygyň noldan ýokary ahmiýeti bar bolsa, onda ähli zat kadaly bolar, ýöne eger uýtgeýan ululygyň ahmiýeti nola deň bolsa ýa-da ýaramaz (otrisatel) ahmiýeti bar bolsa, onda gaýtarylýan ahmiýet kesgitlenen däldir we funksiýa „zir-zibil“ diýlip atlandyrylýan erkin ahmiýeti gaýtarar. Şu ýagdaýda kompilyasiýa edilende duýduryjy habar çykarylýar: "warning Ç4715:sum: ahmiýet ýerine ýetirmegiň käbir ýollarynda gaýtarylmaýar". Şunun ýaly anyk däl ýagdaýlardan gaça durmak üçin funksiýanyň bedeninin ahyrynda operator duýdurmazdan return ahmiýeti goýar (mysal):

```
"it sum (int x, int
y) { if (x > 0) {
    return x + y;
}
return 0;
```

Eger funksiýanyň adynyň önünde void esasy sözi görkezilen bolsa, onda return operatory bolup bilmez. Emma, eger funksiýany ýerine ýetirmegi möhletinden ön togtatmak zerur bolsa, onda return operatory gaýtarylýan ähmiýetsiz görkezilyär. Mysal:

```
void print_ok() {
    std::cout<< "OK"<<std::endl;
    return; // funksiýany möhletinden ön togtatmak
    std::cout<< "bu amal ýerine ýetirilmeýar!!!";
}
```

Programmadan funksiýa çagyrylanda funksiýanyň ady görkezilyär, sondan soň tegelek ýaýlaryň ahmiýetleri berilýär. Eger funksiýa parametrleri kabul etmeýän bolsa, onda dine tegelek ýaýlar görkezilyär. Eger funksiýa ähmiýeti gaýtarýan bolsa, onda ony uýtgeýan ululyga berip ýa-da inkâr edip bolar. Funksiýanyň kesgitlemesinde parametrleriň sány we tipi çagyrylandaky parametrleriň sány we tipi bilen gabat gelmelidir, bolmasa ýalňyslyk barada habar çykarylýar. Üç funksiýany çagyrmagyn mysaly:

```
print ("Message ") ; //funksiýa Message çykaryar
print ok () ;          // parametrsiz funksiýany
çagyrmak
z = sum (10,20);       //z uýtgeýan ululyga ahmiýet
berilýar
```

Berlen ahmiýetler funksiýanyň kesgitlemesinde şol bir orunda (pozisiýada) yerleşen uýtgeýan ululyklara berilýär. Meselem, sum () funksiýasy peýdalanylanda x uýtgeýan ululyga 10 ahmiýet berler, y uýtgeýan ululyga bolsa 20 ahmiýet berler. Funksiýany ýerine ýetirmegiň netijesi uýtgeýan ululyga berilýär. Mysal hökmünde üç dürli funksiýany doredeliň we olary çagyralyň (listing 6.1)

```
#include <iostream>
// funksiýalary yglan etmek
int sum(int x, int y) ; // ýada int sum(int, int);
void print(const char *str) ; // ýada void print(const
char*);
void print ok();          // ýada void print ^ ok (void)
;
int main () {
    // funksiýalary
    çagyrmak
    print("Message");     // Message
    print ok() ;          // OK
    std::cout << sum(10, 20) << std::endl;          // 30
    std:
    :cin.get() ;
    return 0; }
    // funksiýalaryň
    kesgitlemesi int sum (int x, int y) { // iki
    paramametr return x + y; }
    void print (const char*str) { //
    birparametr std::cout <<str <<std: : endl; }
```

```
void print ok() { // parametrsiz
std::cout<<"OK"<<std::
endl; }
```

Funksiýalary yglan etmelerin we esgitlemeleriniň yerleşşi

Programmadaky ähli instruksiýalar yzygiderli ýokardan aşak yerine ýetirilýar. Bu bolsa programmada funksiýany peýdalanmazdan oň ony deslapdan yglan etmelidigini aňladýar. Sonuň üçin hem funksiýany yglan etme funksiýany çagyrmagyň önünde yerleşmelidir. Bir funksiýanyň kesgitlemesini beýlekiniň içinde ýerlesdirip bolmaýandygyna üns beriň. Seýlelikde, funksiýanyň ady hemişe global identifikátor bolup durýar.

Uly bolmadyk programmalarda, funksiýanyň kesgitlemesi main () funksiýasynyň önünde yerleşen ýagdaýynda, funksiýany yglan etmegi görkezmezlige ýol berilýar. Onsoňam main () funksiýasy yglan etmegi talap etmeýar, çünki ol birinji bolup çagyrylýar (mysal). #include <iostream>

```
int sum(int x, int y) { return x + y; }
int main () {
std: : cout<<surat (10, 20) <<std: :
endl; // 30 s t d : : cin . get() ;
return 0; }
```

Funksiýalaryň sany köpelende funksiýanyň içinde ikinji funksiýanyň, ikinjiniň içinde bolsa üçünjiniň çagyrylyan ýagdaýy yuze çykyar. Netijede nâme ön döräpdir - towukmy ýa-da ýumurtga diýen soragy çözmeli bolýar. Şunun ýaly ýagdaýdan gaça durmak maksady bilen, funksiýalaryň yglan edilmegini programmanyň başynda, main () funksiýasynyň önünde, kesgitlemäni bolsa main () funksiýasyndan soň ýerlesdirmek gerek. Şu ýagdaýda funksiýalaryň kesgitlemesiniň yerleş tertibinin ahmiýeti ýokdur (mysal): #include <iostream>

```
int sum (int, int); // yglan
etmek int main() {
std::cout << sum(10, 20) << std::endl;
// 30 s t d : : ç; i n . g e t ( ) ;
return 0; }
int sum(int x, int y) {
//kesgitleme return x + y; }
```

Programmanyň möçberi artanda (giňelende) funksiýalaryň yglan etmeleri we kesgitlemeleri barha köpelyär. Şu ýagdaýda programmany birnäçe aýratyn faýllara bölyärler. Funksiýanyň yglan etmelerini h (kähalatda hpp) giňeltmesi bilen sözbaşy faýlyna çykarýarlar, funksiýalaryň kesgitlemelerini bolsa cpp giňeltmesi bilen sonuň ýaly atlandyrylýan faýlda ýerlesdirýarlar. Faýllaryň ählisi main () funksiýasyny icine alýan esasy faýl bilen bir bukjada ýerlesdirilýar.

Soňra #include direktiwasyynyň kömegi bilen galan ähli faýllarda sözbaşy faýlyna birikdirýarlar. Eger #include direktiwasynda sözbaşy faýlynyň ady burç gömüşi ýaý içinde görkezilyän bolsa, onda faýly gözlemek ulgamlayyn bukjalarda amala asyrylýar. Eger ady goşa dyrnak içinde görkezilen bolsa, onda gözleg ilki başda esasy faýl bolân bukjada, soňra bolsa ulgamlayyn bukjalarda geçirilýär.

6.2. Funksiýalara görkezijiler

Funksiýa bilen iki iş geçirip bolar: ony çagyryp we onuň salgysyny alyp bolar. Funksiýanyň salgysyny almak bilen alnan görkezijini soňra funksiýany çagyrmak üçin peýdalanyp bolar (mysal):

```
void f (int x) { ...  
}  
void (*pf) (int) ;           // funksiýa görkeziji(ýaýlar hökmandyr)  
void g()  
1 pf&f;                     // pfffunksiýa görkezyär  
pf(0) ;                     // pf görkezij i arkaly f funksiýany çagyrmak
```

Kompilýator pf görkezij i bolup durýandygy tanaýar we onuň görkezyän funksiýasyny çagyýar. Yagny, funksiýanyň görkezijisini *amaly kömegi bilen atlandyrmak hökman däl. Edil sonun ýaly, funksiýanyň salgysyny almak üçin & amalyndan peýdalanmak hökman däl.

Funksiýanyň görkezijileri funksiýalaryň özlerinin parametrleri ýaly yglan edilýar. Berlen wagtynda funksiýalaryň tipleri takyk gabat gelmelidir (mysal).

```
typedef void                // funksiýa görkejini yglan etmek üçin typedef  
(* p F) (i n t) ;         Tipiň ulanylyşy  
  
PF pf;                     // funsiýa görkezij inin özüni yglan etmek  
void f 1(int x) {  
  
int f2(int x) {  
void f3(char x) {  
  
void f ()  
{ pf = &f1;                // dogry  
  pf = &f2;                 // ýalňys  
  p f = & f 3 ;            // ýalňys  
}
```

Görkezijiniň üstünden funksiýalar çagyrylanda hem parametrleri bermeginiň kadalary funksiýalaryň göni dä! çagyrylandakysy ýalydyr.

7. KLASLAR WE OBÝEKTLER BILEN İŞLEMEK. PROGRAMMANYŇ KODUNDA DÜŞÜNDİRİŞ BELLİKLERİNİ GOÝMAK (COMMENTS). GAÝTADAN YÜKLEMEK. YALNYŞLYKLARY DOLANDYRMAK

7.1. Obýekte gözükdirilen programmirlleme (OGP)

Obýekte gözükdirilen programmirlleme (OGP) - bu şol bir kódy birnäçe gezek peýdalanmaga mümkinçilik berýan programmany guramagyň usulydyr. Funksiýalardan tapawutlylykda OGP dine bir programmany fragmentlere bölmäge mümkinçilik bermän, eýsem gündelik durmuşdaky predmetleri obýektler görnüşinde beýan etmäge, şeýle hem bu obýektleriň arasynda aragatnaşygy guramaga mümkinçilik beryär.

Esasy **düşünjeler**

OGP-de esasy "kerpiç" klas (dereje) bolup durýar. Klas uýtgeýan ululyklaryň we bu uýtgeýan ululyklary dolandyrmak üçin funksiýalaryň toplumyny öz içine alýar. Uýtgeýan ululyklar atributlar (hasiýetler, uýtgeýan ululyklar-agzalar ýa-da maglumatlar-agzalar), funksiýalar bolsa usullar (ýa-da funksiýalar-agzalar) diýlip atlandyrylýar. Jemlenen ýagdaýda atributlara we usullara klasyň agzalary diýilýar. Klas döredilenden soň onuň ady maglumatlaryň täze tipi bolýar. Seýlelikde, peýdalanyjylar klasy C++ dilinin mümkinçiliklerini giňeldýar.

Klasyň esasynda köp sanly obýektleri döredip bolar. Şunda her bir obýekt üçin lokal uýtgeýan ululyklaryň öz toplumy döredilýär. Mysal üçin, setirler öwrenilende biz string klasyna seretdik. Uýtgeýan ululyk ygılan edilende klasyň ady maglumatlar tipinin yerine görkezilipdi (mysal):

```
std: :string str("string1") ,str2("string2") ;
```

Bu mysalda iki obýekt - str 1 we str2 obýektleri kesgitlendi, olar string klasynyň ekzemplýarlary bolup durýarlar. Her bir obýekt inisializasiýada görkezilen öz setiri baradaky maglumatlary saklaýar. Bir obýektiň uýtgemegi beýleki obýektiň ähmiýetine täsir etmeyär. Her bir obýektiň içinde, biz üçin mâlim bolmadyk ýagdaýda, ýadyň setirini saklamak üçin zerur bolan möçberini bölüp bermek, artdyrmak we boşatmak boyunça amallar geçirilýar. Seýlelikde, obýekt döredilende ýa-da ol üýtgedilende bize ýadyň anyk möçberini görkezmek we ýadyň boşamagyna gözegçilik etmek zerur daldır.

Setir bilen iş geçirmek mümkinçiligi bolar ýaly, string klasy köp usullary beryär. Mysal üçin, size () usulynyň kömegi bilen biz setirdäki simwollaryň sanyny aldyk. Usul çagyrylanda onuň ady obýektiň adyndan soň nokadyň üsti bilen görkezilýär (mysal).

```
std: : cout << s t r 1 . s i z e () << s t d : : e n d J;
```

Usullardan başga-da, *string* klasy käbir operatorlary gaýtadan işe girizýär. Meselem, konkatenasiýa üçin + operatory peýdalanylýar (mysal).

```
str1=str1+str2
```

Operatorlary gaýtadan işe girizmek yörite atlary bolan usullaryň kömegi bilen geçirilyar. Klasda "operator" usulyňy kesgitlemek bilen operatoryň özüni alyn barşyny öz garaýsyňa göre üýtgedip bolar. Mysal üçin, + operatory gaýtadan işe girizip gosmagyň ýerine aýyrmagy geçirip bolar. Emma klasyň peýdalanyjylary şunun ýaly başlangyja oňyn baha bermezler. Şeýle-de bolsa, käbir ýagdaýlarda operatoryň manysynyň üýtgedilmegi örän peýdaly bolup biler. Mysal üçin, cout klasy konsol penjiresine maglumatlary çykarmak üçin bit boyunça süýşüryan « operatory gaýtadan işe girizýär.

Gündelik durmuşdaky menzeş ýagdaýlar bilen deneşdirilende telewizor klasyň ekzemplýary (obýekti), telewizoryň döredilen taslamasy bolsa klas bolup durýar diýip bolar. Şu taslama boyunça köp telewizor (klasyň köp ekzemplýaryny) döredip bolar. Telewizoryň korpusyndaky duwmeler we öýjükler, korpusyň özi hem degişli klasyň aýry-aýry obýektleri bolup durýar. Telewizoryň işledilmegine ýa-da öçürilmegine, kanallara geçirilmegine getirýän duwmäni basmagyň özi usullaryň kömegi bilen ýerine ýetirilýar. Duwmaniň häzirki ýagdaýyny, şeýle hem duwmäniň ölçegini, onuň reňkini we beýlekileri saklamak üçin atributlar niýetlenendir. Şol bir düwme dürli heketleri ýerine ýetirip bilýar. Mysal üçin, duwmä bir gezek basylanda bir hereket ýerine ýetirilýar, baş sekundyň dowamynda duwmä basyp durmak (gaýtadan işe girizmek usuly) bolsa düybünden başga hereketiň ýerine ýetirilmegine getirýar.

Telewizordaky bar zatlaryň ählisi kopusyň içinde bolýar, olar görünmeyer. Telewizordan peýdalanmak üçin onuň icinin nähili gurlandygyny bilmek asla gerek daldır, duwmaniň (elýeterlilik interfeýsi) we peýdalanyjy üçin gollanmanyň (klasa degişli resminamalar) bolmagy ýeterlikdir. Edil sonuň ýaly, klasy işlap taýýarlaýjy elýeterlilik interfeýsini berip, koduň galan bölegini bolsa uýtgemelerden gorap biler (OGP-de maglumatlary gizlemek inkapsulýasiýa diýlip atlandyrylýar). Mundan beýlak klasy işlap taýýarlaýjynyň, elýeterlilik interfeýsini uýtgetmezden, klasyň içinde uýtgetmeleri geçirmek mümkinçiligi bardyr. Zerur bolan ýagdaýynda peýdalanyjy klasy uýtgetmage synanyşyp biler. Ony telewizoryň arka yüzündäki gapagy aýyrmak we onuň içinde nämedip bir zatlar etjek bolmak bilen deneşdirip bolar. Bu hereketiň netijesi ýerine ýeririjiniň hünar derejesine baglydyr. Eger ol ussa bolsa, onda telewizoryň gowy işlemegi mümkindir, eger yöne bir gyzyklanýan adam bolsa, onda täze telewizor satyn almaly bolaýmagy ähtimaldyr.

Bir ýagdaýda telewizory işlap taýýarlaýjy korpusyň içinde DVD-pleýer goşmak bilen, täze modeli çykarmagy karar edýar. Telewizoryň icindäki bloklaryň köpüsi önki modeldäki ýaly bolup galar. Esasy uýtgemeler korpusa degişli bolar. Şeýlelikde, DVD-pleýerli telewizor önki modelin konstruksiýasyny miras alýar, täze funksionallyk goşyar (DVD-pleýer) we käbir usullary gaýtadan kesgitleýar (korpusy uýtgedýar).

Miras almagyň netijesinde döredilen klasda haýsydyr bir usulyň özüni alyp barşy üýtgap biler. Mysal üçin, awtobis hem-de zynjyrly traktor özünde hereketiň usulyňy beýan edýan usul kesgitlenen awtomobil klasynyň mirasdüşerleri bolup durýarlar. Awtobusyň hereketiniň zynjyrly traktoryň hereketinden tapawutlanýandygy aýdyňdyr. Sonuň üçin hem awtobus klasy usuly awtomobil

klasyndan ony gaýtadan kesgitlemän miras alyp biler, zynjyrly traktor bolsa usuly gaýtadan kesgitlemeli bolar.

Indi bu klaslaryň ählisine adybir usul elýeter bolýar, ýöne bu usulyň durmuşa geçirilişi tapawutlanýar. Seýle-de bolsa, bu usul şol bir bereketi - ulag serisdeleriniň hereketini ýerine ýetirýar. OGP-de şunun ýaly ýagdaý poliformizm diýip atlandyrylýar.

Şeýlelikde biz OGP-iň esasy üç düşüňjesi bilen tanyşdyk:

- inkansulýasiýa - maglumatlary klasyň içinde gizlemek;
- miras almak - binýatlyk alasyň esasynda emele getirilýän klaslary döretmek mümkinçiligi. Şunda emele getirilen klas awtomatik usulda binýatlyk klasyň mümkinçiliklerini alýar we täze funksionallygy goşup ýa-da käbir usullary gaýtadan kesgitlep biler;
- poliformizm - adybir usulyň ýerine ýetirýän hereketiniň manysy bu hereketleriň ýerine ýetirilýän obýektine baglydyr.

Haýsy zat klas görnüşinde, haýsy zatlar bolsa usullar ýa-da atributlar görnüşinde görkezilip bilner? Eger söz has at bolup durýan bolsa (awtomobil, telewizor, duwme we ş.m.), onda ol klas hökmünde beýan edilip bilner. Usul obýektiň uýtgeýsini beýan edýar, mysal üçin, awtomobil hereket edip başlady, üzüksiz hereket edýar, saklandy. Atribut obýektiň we onuň hasiýetnamalarynyň häzirki ýagdaýyny, mysal üçin, duwmäniň ölçegini we onuň reňkini, basmak alamatyny saklamak üçin niýetlenendir.

Bu gysgaça syny jemlemek bilen, OGP-iň goldawynyň C++ dilini dine programmirlemegiň prosedura stili goldaýan C dilinden tapawutlandyrýar. C# we Java dillerinden tapawutlylykda programmirlemegiň OGP stili C++ dilinde hökmany stil bolup durmaýar. Siz öz programmalaryňyzda dine prosedura stihni peýdalanyň bilersiniz. Haýsy stiliň peýdalanylmalydygy anyk ýagdaýa bagly bolup durýar. Ýöne siziň programmaňyz näçe uly bolsa, OGP stiliň artykmaçlyklary aýdyň bolar.

Klas *class* esasy sözün kömegi bilen şu shema boyunça yglan edilyär:

```
dass [<klasyň ady>] {  
<klasyň ýapyk agzalaryny yglan etmek>; public:  
<klasyň açyk agzalaryny yglan etmek>; private:  
<klasyň ýapyk agzalaryny yglan etmek>; protected:  
<klasyň goralan agzalaryny yglan etmek>; }  
[<i>iiýtgeýän ululyklary yglan etmek>]
```

Class esasy sözden soň klasyň ady berilýar, ol maglumatlaryň täze tipi bolýar. Klasyň ady ýol berilýän identifikátor bolmalydyr, oňa uýtgeýän ululyklaryň atlaryna bildirilýän talaplar ýaly talaplar bildirilýar. Eger ýapýan şekilli ýaýdan soň uýtgeýän ululyk görkezilen bolsa, klasyň adyny bermän hem bolar. Klasy yglan etmegiň sonunda nokatly otur hökmany bolup durýar.

Şekilli ýaýlaryň içinde klasyň agzalary - atributlar we usullar yglan edilyär. Yglan etmeden ön elýeterliligiň spesifikatorynyň adyny görkezip bolar:

- *public* - açyk. Identifikátor daşarky peýdalanyjy üçin elýeterdir,

■ *private* - ýapyk. Identifikátor dine şu klasyň icinde elýeterdir,

■ *protected* -- goralan. Identifikátor daşarky peýdalanyjy üçin elýeter däl, ýöne şu klas üçin we bu klasyň nesilleri üçin elýeterdir. Üns berin!

Eger elýeterlilik spesifikatory görkezilmedik bolsa, onda idenifikator ýapyk bolup durýar.

Elýeterlilik spesifikatorlary klasyň atributlarynyň ahmiýetine gözegçilik etmek üçin, şeýle hem dine klasyň durmuşa geçirmek üçin niýetlenen usullarynyň peýdalanylmagyny gadagan etmek üçin niýetlenendir. Mysal üçin, eger atributda belli bir ahmiýetleri saklamak göz önünde tutulan bolsa, onda ähmiýet berilmezinden ön biz ahmiýetiň käbir sertlere laýyk gelýändigini barlar bileris.

Eger islendik peýdalanyjyda biziň barlagymyz ýaly barlag geçirmän, islendik zady girizmek mümkinçiligi bar bolsa, onda hiç hili gözegçilik barada güiTün bolman biler. Şunun ýaly gizlemek konsepsiýasy inkapsullaşdyrma diýlip atlandyrylýar.

Klasyň yglan edilmegi uýtgeýan ululygy kesgitlemän, dine maglumatlaryň täze tipini beýan edýar, sonun üçin hem oňa ýat bölünip berilmeýar. Uýtgeýan ululygy yglan etmek üçin onuň ady ýapýan şekilli ýaýdan soň, klas yglan edilende ýa-da aýratynlykda, klasyň adyny maglumatlaryň tipi hökmünde peýdalanmak bilen görkezilýär (mysal).

```
<klasyň ady> <uýtgeýan ululygyň ady >; /şoJ. bir wagtyň  
özünde açyk atributly klasyň we uýtgeýan jlugyň yglan  
etmegin mysaly
```

```
i; S S PoInt {  
.ç ]. j ç :  
    i n t x  
    ; i, n t  
    y ; }  
    point  
    1;
```

Uýtgeýan ululygy aýratynlykda yglan etmegiň mysaly:
Point point 2 ;

Public spesifikatorynyň bardygyna üns beriň. Eger bu spesifikator görkezilmedik bolsa, onda x we y atributlaryna elýeterliliği alyp bolmaz, çünki klaslaryň agzalary duýdurmazdan ýapyk bolup durýar we dine klasyň icinde elýeter bolýar.

Spesifikatoryň hereketi nobatdaky spesifikatora çenli ýa-da ýapýan şekilli ýaýa çenli dowam edýar. Klas yglan edilende birnäçe birmenzeş spesifikator bolup biler. Mysal:

```
Lass Point { public:  
    int x;  
public:  
    int y;  
    } point 1;
```

Usullaryň ahlisiniň klasyň içinde durmuşa geçirilýän sertinde klaslaryň kesgitlemesini funksiýanyň ýa-da beýleki bir klasyň icine salyp bolar. Şu ýadaýda **klas** dine funksiýanyň içinde görünyar. Mysal:

```
func() {
    class Point { public:
        int x ;
        int y ;
    } point;
    point.x =10;
    point.y = 40;
    std::cout<<point.x<<std::endl; // 10
    std::cout<<point.y<<std::endl; //40 }
```

Bir klasyň içinde beýleki klasyň obýektlerini peýdalanmaga ýol berilýar. Mysal hökmünde Point (nokat) klasyny yglan edýaris, soňra ony Reçt (gönüburçlyk) klasynyň içinde gönüburçlygyn koordinatyny beýan etmek üçin peýdalanarys (mysal):

Klaslary ulanmagyň mysaly (Listing 7.1)

```
#include <iostream>
using namespace std;
class Point {
public:
    int x;
    int y;
};
class Rect {
public:
    Point top_left;
    Point
    bottom_right;
};

int main ()
{
    Rect
    rect;
    rect.top_left.x = 0; rect.top_left.y = 0;
    rect.bottom_right.x = 100; rect.bottom_right.y =
    100; std::cout << rect.top_left.x<<" "
        <<rect.top_left.y<<std::endl
        <<rect.bottom_right.x<<" "
        <<rect.bottom_right.y<<std::
    : endl ; std::cin.get () ; return
    0;
```

7.2. Atributlary we usullary yglan etmek

Atributlary yglan etmek

Klasyň içinde atributlary yglan etmek adaty uýtgeýän ululygy yglan etmek ýaly geçirilýar, dine maglumatlar tipinin önünden saklayyş spesifikasiýalaryny görkezmeli, şeýle hem atributyň inisiallaşdyrmagy geçirmeli däl. Klasyň

ekzemplarynyň üstünden atributyň ahmiýetini takyk notasiýanyň kömegi bilen berip ýa-da alyp bolar.

```
<iýtgeýan ululyk>.<atributyň ady> =  
<ahmiýeti> <ahmiýeti> = < uýtgeýan ululyk  
>.< atributyň ady >
```

Mysal üçin:

```
point 1.x = 10;  
point 1. y = 40;  
sr.d::cout << point1.x << std::endl; // 10  
std::cout << point1.y << std::endl; // 40
```

C++ dilinde strukturanyň klasslaryňka menzeş funksionaly bardyr. Bir tapawudy strukturanyň agzalarynyň elýeterliliginiň açyk bolup durýandygyndan, klasyň agzalarynyň elýeterliliginiň bolsa, duýdurmazdan, ýapyk bolup durýandygyndan ybaratdyr. Iki ýagdaýda hem duýdurmazdan özüni alyp barşy uýtgedip bolar. Point strukturasyny ygylan etmek şunun ýaly bolýar (mysal):

```
struct Point { // strukturany we uýtgeýan ululygy ygylan etmek  
    int x ;  
    int y;  
  
    }point;
```

Şeýle hem strukturalarda bolşy ýaly, klasyň icinde bit meýdanlaryny peýdalanmaga ýol berilýar. Bit meýdanlary bir uýtgeýan ululykda bitlerin görkezilen sanyny tutýan birnäçe ahmiýetleri saklamaga mümkinçilik bermek bilen, aýry-aýry bitleri elýeter edýar. Bit meýdanyny ygylan etmegiň şunun ýaly görnüşi bolýar (mysal): <máglumatyň tipi> [<meýdanyň ady>]:<bitde uzynlygy>;

Bir klasda bir wagtyň özünde bit meýdanlaryny we adaty meýdanlary peýdalanyp bolýar. Bit meýdanynyň adyny görkezmän hem bolýandygyna üns beriň, mundan başga-da, eger meýdanyň uzynlygy bir bite deň bolsa, onda meýdanyň adyndan ön unsigned esasy söz görkezilmelidir. Bit meýdanlary bolan klasy ygylan etmek (mysal):

```
: '. ass  
Status {  
public:  
    unsigned :3;  
    unsigned a:1 ;  
    unsigned b:1;  
    unsigned  
    ç:1; status;
```

Bit meýdanynyň elýeterliligi klasda atributlaryň elýeterliligii ýaly amala asyrylýar. s t a t u s . a = 1 ; Status.b=0; Status.ç=1

Usullary ygylan etmek we kesgitlemek

Klasyň içinde usullary yglan etmek adaty funksiýanyň yglan edilişi ýaly amala asyrylýar. Usulyň kesgitlemesini adaty klasyň daşynda ýerleşdirýarlar, şunda klasyň adyny usulyň adyndan oň görkezýärler. Iki adyň arasynda : : operatory goýulýar. Klasyň usullaryna ýüzlenilende şu biçuw peýdalanylýar (mysal):

```
<klasyň ekzemplýary>.<usulyň ady>(parametrler)
```

Usulyň kesgitlemesini klasyň yglan etmegiň içinde yerleşdirip bolar. Şu ýagdaýda usulyň icindäkiler usuly çagyrmagyň ýerine goýulýar. Sular ýaly usullar *icindaki usullar* diýlip atlandyrylýar. Icindaki usullar bilen uly bolmadyk ähli usullary geçirip bolar. Usullary yglan etmegiň we kesgitlemegiň mysaly 7.2 listingde getirýär.

```
Usullary yglan etmek we kesgitlemek (Listing 7.2)
#include <iostream>
class C {
    int x_; // Ýapyk atribut
public :
    void set_x(int y) { x_=y //Icindaki usuly kesgitlemek
    int get_x (); // Adaty usuly yglan etmek } obj ;

int main() {
    // ýapyk atributyň ahmiýetini uýtgetmek
    obj.set_x(10);
    // ahmiýet almak
    std::cout << obj.get_x () << std::endl; // 10
    std::cin.get();
    return 0 ;

int C::get_x () { // adaty usuly kesgitlemek
return x ;
```

7.3. Konstruktorlar we destruktörler

Klasy ekzemplýary döredilende ilki başky ahmiýetleri haýsydyr bir atributlara bermek üçin klasyň ady ýaly ady bolan usul döredilmelidir. Gaýtarylýan ahmiýetiň tipi görkezilmeyär. Şunun ýaly usul konstruktör diýlip atlandyrylýar. Obýekt döredilende konstruktör hemişe awtomatik usulda çagyrylyar.

Konstruktoryň parametrleriň tipi ýa-da olaryň sany bilen tapawutlanýan tadan işe girizilen wersiýalary bolup biler. Eger klasyň içinde konstruktör ýok bolsa, onda parametrleri bolmadyk konstruktör awtomatik usulda döredilyär. Şu ýagdaýda obýekt şunun ýaly yglan edilýar (mysal):
 klasyň ady> <uýtgeýan ululygyň ady>; *ýa-da* şunun ýaly döredilyär: (mysal) klasyň ady> <uýtgeýan ululygyň ady>=<klasyň ady>

Eger klasyň içinde peýdalanyjy konstruktory yglan edilen bolsa, onda konstruktör duýdurmazdan döredilmeyär. Bu eger siz bir parametrli konstruktör

döreden bolsaňyz, onda obýekt döredilende parametrleri görkezmeli boljakdygyny aňladýar. Parametrleri görkezmän obýekti döretmek mümkinçiligini almak üçin paramsiz konstruktör goşmaça döredilmelidir. Parametrleri görkezilen obýekti yglan etmek şunun ýaly görnüşde bolar (mysal):

```
<klasyň ady> <uýtgeýan ululygyň ady> (<parametri> (<parametr1> [...,  
<parametrN>] ); Şeýle hem obýekti yglan etmegiň alternatiw warianty  
bardyr (mysal):  
<klasyň ady> <uýtgeýan ululygyň ady>=<klasyň ady>  
    (<parametr1> (<parametr1> [, . .,<parametrN>] ) ) ;
```

Birnäçe konstruktoiriary bolan klasy döretmek we klasyň ekzemplýaryny döretmegiň dürli usullary 7.3 listingde görkezilýär.

Obýekti döretmegiň usullary we konstruktorlary gaýtadan işe girizmek (Listing7.3).

```
#include <iostream>

class C {
    int x_, y_;
public:
    C() {x_ = 0; y_ = 0; };
    C(int x) {x_ = x; y_ = 0;}
    C(int x, int y) {x_ = x; y_ = y;}
    int get_x() {return x_;}
}

int main() {
    C obj1;
    C obj2 = C();
    C obj3(10);
    C obj4 = 20;
    C obj5 = C(30);
    C obj6(40,30);
    C obj7 = C(50,5);

    //C obj8(); // Ýalňyşlyk;
    std::cout << obj1.get_x() << std::endl;    // 0
    std::cout << obj2.get_x() << std::endl;    // 0
    std::cout << obj3.get_x() << std::endl;    // 10
    std::cout << obj4.get_x() << std::endl;    // 20
    std::cout << obj5.get_x() << std::endl;    // 30
    std::cout << obj6.get_x() << std::endl;    // 40
    std::cout << obj7.get_x() << std::endl;    // 50
    std::cin.get();
    return 0;
}
```

```
}
```

Mysaldan görnüşi (obj4 obýektiniň döredilişine seret), eger konstruktör dine bir parametri kabul edýan bol sa, onda klasyň ekzemplýaryny döretmegin şunun ýaly usuly mümkin bolar (mysal):

Şu ýagdaýda ahmiýetiň klas obýektine açyk däl öwriilmegi bolup geçýar. Şunun ýaly owriilmegiň önüni almak üçin bir parametri kabul edýan konstruktoryň önünden explicit esasy söz görkezmelidir.

```
include <iostream>

class {
    int x , y ; public:
explicit C(int x) { x_=x; y_=0; return x_ ; } } ;

it wain () {
    objí (10) ;
```

Konstruktory we destruktory çagyrmagyň tertibi (Listing 7.4) (mysal):

```
#include <iostream>
class C {
{ int x_;
public:
C (int x) { // konstruktor
    x_ = x ;
    std::cout << "C (int x) " << x_ << std::endl;
}

~C() {

    std :: cout << „C(int x“ << x_ << std :: endl;

}

}

obj1(1), obj2(2);

int main() {
C obj3(3);
std :: cout << „main()“ << std :: endl;
return 0;
}
```

Ýerine ýetmegiň netijesi:

```
C(int x) 1
C(int x) 2
C(int x) 3
main()
```

~C() 3
~C() 2
~C() 1

Netijeden görnüşi ýaly, global obýektleriň konstruktorlary (klasy ygılan etmeden soň yerleşendir) dolandyryş main () funksiýaýna berilmezinden ön yerine ýetirilýar. Lokal obýektleriň konstruktorlary funksiýanyň icindaki obýektleri ygılan etmek tertibinde çagyrylyar. Destruktorlary çagyrmak garşylykly (tersine) tertipde geçirilyar. **7.4. Ýalňyslyklary dolandyrmak**

1. Ýalňyslary işlap geçmek

Programmanyň dürli modullardan konstruktirlenýan mahalynda we aýratyn hem bu modullaryň garaşsyz işlenip taýýarlanan kitaphanalarda bolýan wagtynda, ýalňyslary işlap geçmek iki bölege bölünip biler:

- a) lokal derejede çözülip bilinjek ýalňys ýagdaýyň yüze çykandygy barada maglumaty emele getirmek;
- b) beýleki yerlerde yüze çykarylan ýalňyslary işlap geçmek.

Kitaphananyň awtory ýalňyslygy yerine ýetirilýan wagtynda yüze çykaryp biler, yöne düzgün bolşy ýaly, şu ýagdaýda ol nâme etmelidigini bilmeýar. Kitaphanadan peýdalanyjy ýalňyslar yüze çykanda nâme etmelidigini bilip biler, yöne olary yüze çykaryp bilmeýar, tersine bolan ýagdaýynda ýalňyslar peýdalanyjynyň kodunda işlenip geçilerdi, olary gözlemek bolsa kitaphana tabşyrylardy. Şunun ýaly problemalary çözmekde kömek etmek üçin kadadan çykmalar düşünjesi girizildi.

Fundamental ideýa problemany yüze çykaran, yöne ony nähili çözmelidigini bilmeýan funksiýanyň ony çagyran (gönüden-göni ýa-da gytaklaýyn) funksiýa yüze çykan problemany çözüp biler diýen umyt bilen kadadan çykmalary emele getirýandiginden ybaratdyr. Şu tipdäki problemany çözmek isleyän funksiýa özüniň sular ýaly kadadan çykmalary tutýandygyny (önüni tutýandygyny) görkezip biler.

Ýalňyslary işlap geçmegin şunun ýaly stili dăp bolan tehnikalaryň köpüsine garanynda ileri tutulýar. Lokal çözülip bilinmeyän problēma yüze çykarylanda funksiýa:

1. yerine ýetirmegi bes edip;
2. "ýalňysy" aňladýan ahmiýeti gaýtaryp;
3. ýol berilýan ahmiýeti gaýtaryp we programmany kadaly dăl ýagdaýda goýup biler.

1-nji wariant - "ýerine ýetirmegi bes etmek" - bu kadadan çykmalaryň tutulnmaýan mahalynda, duýdurmazdan bolup geçýan ýagdaýdyr. Ýalňyslaryň köpüsi üçin nâmedir bir gowy zat oýlap tapmalydyrys. Kitaphana, gürrünsiz tamamlajy yerine yetirme, oňa bildirilýan birinji talap ygtybarlylyk bolup durýar.

2-nji wariant - "ýalňyslyk barada duýdurş berýan ahmiýeti gaýtarmak" - muny hemişe yerine ýetirip bolmaýar, sebäbi köplenç kabul ederlikli ahmiýeti ýokdur. Şunun ýaly çemeleşmanin ulanarlykly wagtynda hem ol amatly dăldir,

sebäbi her bir çagyryşyn netijesi ýalňys ahmiýet babatynda barianmalydyr. Bu programmanyň möçberini has-da artdyryp biler.

3-nji wariantyň - "ýol berilýan ahmiýeti gaýtaryp we programmany kadaly däl ýagdaýda goýmak" - çagyryan funksiýanyň programmanyň kadaly däl ýagdaýdadygyny bilip bilmezligi ýaly ýetmezçiligi bardyr.

Kadadan çykmalary işlap geçmegin mehanizmi döp bolan usullaryň yeterlik däl. elegant däl we ýalňyslygyň täsirine düşüp bilýan halatlarynda bu usullara alternatiwany beryär. Ol ýalňyslary işlap geçmek koduny "adaty" koddan açyk bölüp aýyrmak usuly bolup durýar, şeýle etmek bilen ol programmany gowy okap bolar ýaly we dürli instrumental serişdeler üçin has laýyk geler ýaly edýar. Kadadan çykmalary işlap geçmek mehanizmi ýalňyslary işlap geçmegin has yzygiderli usulyny beryär, netijede koduň aýratynlykda ýazylan fragmentleriniň arasynda özara bereketi ýeňillesdirýar.

Ýalňyslary işlap geçmegin çylşyrymly wezipe bolup durýandygyna we kadadan çykmalary işlap geçmegin mehanizminiň (alternativ usullara garanynda ýokary derejede formai laşandygyna garamazdan) ýerine ýetirmegi lokal dolandyrmagy üpjün edýan diliň serişdeleri bilen deneşdirilende struktirlenişinin belli bir derejede pesdigine düşünmek gerek. C++ kadadan çykmalary işlap geçmek mehanizmi programmaça ulgamyň şu gurluşynda ýalňyslary has tebigy ýagdaýda işlap geçip bolýan yerinde olary işlap geçmegin serişdesini beryär. Kadadan çykmalar ýalňyslary işlap geçmegin çylşyrymlydygyny has aýdyň edýar. Emma kadadan çykmalar bu çylşyrymlylygyn sebäbi bolup durmaýar.

Kadadan çykmalary işlap geçmegin mehanizmine funksiýadan gaýtarmak mehanizminiň alternatiwasy hökmünde seredip bolar. Sonuň üçin hem ýalňyslara hiç hili gatnaşygy bolmadyk kadadan çykmalary peýdalanmak doly derejede xanunydyr. Emma kadadan çykmalary işlap geçmek mehanizminiň ilkibaşky maksady ýalňyslary işlap geçmek we ýalňyslar yuze çykanda durnuklylygy üpjün etmek bolup durýar.

2. Kadadan çykmalary emele getirmek we tutmak

Aýratyn ýagdaýlary işlap geçmek dolandyryşy we maglumatlary şu tipli gdaýy işlemege isleg bildirilen yerdäki näbelli nokada bermegiň usulyny üpjün edýar. islendik tipiň ýagdaýlary döräp (throw) we tutulyp (catch) bilner we t'unksiýa üçin onda döräp biljek köp ýagdaýlar geçirilip bilner.

Reaksiýa dine gözegçilikli blogyň icinde dörän aňlatmanyň ýerine ýetirilen > agdaýynda ýa-da bu blokdan çagyrylan funksiýalarda peýda bolup biler.

Gözegçilikli blogyň sintaksisi: (mysal) *try {... } <reaksiýalaryň sanawy>*

Reaksiýalaryň sanawy şu konstruksiýadan ybaratdyr: (mysal)

catch (<ýagdaýy yglan etmek>) { ... } [catch (<ýagdaýy yglan etmek >) { ... } ...]

Yagdaý döredilende (ýagny, throw operatory ýerine ýeririlende) dolandyryş reaksiýa berilýar. throw operatorynyň operandasynyň tipi haýsy reaksiýalaryň şu yagdaý\ tutup biljekdigini kesgitleýar.

Eger gözegçilikli blogyň reaksiýalarynyň arasynda laýyk gelýan reaksiýa tapylmadyk bolsa laýyk gelýan reaksiýanyň gözlegi şu gözegçilikli blogy icine alýan gözegçilikli blokda dowam edýar.

Eger programmada laýyk gelýan reaksiýa tapylmadyk bolsa, terminate () funksiýasy çagyrylýar. terminate () funksiýasy set_terminate () funksiýasyna soňky _ezek ýuzlenilende berlen funksiýany çagyryar. terminate () funksiýasyndan ; i _.: rylýan funksiýa, duýdurmazdan, abort () bolup durýar. terminate () funksiýasy tarapyndan çagyrylýan funksiýa programmanyň ýerine ýetirilmegini tamamlamalydyr. Kadadan çykma käbir klasyň obýekti bolup durýar, ol aýratyn ýagdaýy görkezilmegidir. Yalňyşy yüze çykaran kod throw intruksiýalarynyň obýektini emele getirýar. Kodyň fragmenti catch intruksiýasynyň kömegi bilen kadadan çykmany işlap geçmage isleg bildirýar. Kadadan çykmany throw intruksiýa bilen emele getirmegin netijesi kadadan çykmany emele getiren funksiýany gönüden-göni ýa-da gytaklaýyn çagyran funksiýa laýyk gelýan catch yüze çykarylyança stegi çözmek bolup durýar.

Kadadan çykmany emele getirmegin özi ýalňyslyk we onun tipi barada habar beryär. Mundan başga-da, kadadan çykma goşmaça maglumat hem bolup biler. Kadadan çykmalary işlap geçmek tehnikasynyň duýp maksady problēma yüze çykandan soň dikeltmek üçin maglumatlary bermek we ony ygtybarly we amatly usul bilen amala aşyrmak bolup durýar.

Mysa I 1:

```
FILE *open(char *fname)
    FILE *f == fopen (fname,
        "r");
    if(!f) throw fname;
    return f;

void main
    ()
try
    FILE *f1 = open("ini.txt");
    FILE *f2 =
    open("in2.txt"); }
catch (char *str)
    printf("Impossible to open file '%s'!\n", str);
    return;
}
...
}
```

MysaI 2:

```
class
Exl
{privat
e:
int reason;
public: Exl (int r) : reason(r){}
```

```

        int reason() { return reason; }

class Ex2 { };
void f1()
{
    if (...) throw Ex1 (0);
    if (...) throw Ex1 (2) ;

    if (...) throw Ex2
    () ;
}

void f2 ()
{ ...
    if (...) throw Ex2 ( ) ;

void main()
{ try
    {
        f1() ;

        f2() ;

    }
    catch (Ex1 ex) {
        switch (ex.Reason())
        {
        case 0
        case 1
        case 2
        }
    }
    catch (Ex2 ex]
    { ... }

```

3. Kadadan çykmalary tutmak

Mysala seredeliň.

```

try
{ throw E();
} catch (H)
{ ... }

```

Işlap geçiji şu ýagdaýda çagyrylar, eger:

1. E tipdäki ýaly H bolanda;
2. E üçin H anyk elýeter köpçülikleýin binýatlyk klas bolup durýan bolsa;
3. H we E görkeziji bolup durýan bolsalar we 1-nji ýa-da 2-nji olaryň

- salgylanýan tipleri üçin ýerine ýetirilýan bolsa;
4. H salgylanma bolup durýan bolsa we 1-nji ýa-da 2-nji H-nyň salgylanýan tipi üçin ýerine ýetirilýan bolsa.

4. Kadadan çykmalary topara jemlemek

Köplenç ýagdaýda kadadan çykmalar tebigy görnüşde maşgalalara bölünýar. Şundan miras almagyň kadadan çykmalary struktirlmek üçin peýdaly bolup biljekdigi we olar işlenip geçilende kömek edip biljekdigi gelip çykyar. Mysal üçin, matemātiķi kitaphana üçin kadadan çykmalary şunun ýaly görnüşde gurap bolar (mysal).

```
class MathErr { ... } ;  
class Overflow      : public MathErr { ... } ; //ýokardan doldurma  
class Underflow     : public MathErr { ... } ; //aşakdan doldurma  
class ZeroDivision : public MathErr { ... } ; //nola bölmek
```

Bu bize MathErrden gelip çykyan islendik klasýň kadadan çykmalaryny, nähili kadadan çykmanyn yüze çykandygy barada alada etmezden, işlap geçmage mümkinçilik beryär (mysal). try

```
{ ... }  
catch (Overflow)           // Overflow kadadan çykmany i släp geçmek  
{ ... }  
  
catch(MathErr)             // MathErr, kadadan çykmany işlap geçmek  
{ ... }
```

Kadadan çykmalary basgançaklar (ierarhiýa) görnüşinde guramagyň koduň ygtybarlylygy üçin uly ähmiyeti bardyr. Matemātiķi kitaphana täze kadadan çykma girizilende matemātiķi kadadan çykmalaryň ählisini isläp geçmage synanyşýan koduň her bir fragmenti modifikasiýa geçirilmegine degişli bolup biler. Bu köp zahmetin sarp edilmegini talap edýar we umuman alanyňda mümkin dälendir.

5. Gaýtadan emele getirmek

Kadadan çykmany tutmak bilen, işlap geçiji ýalňyslygy doly isläp geçip bilmeyändigini barada karara gelip biler. Şu ýagdaýda isläp geçiji başaran zadyny edýar, sondan soň kadadan çykmany tazedem emele getirýär (mysal). try

```
{ ... }  
catch (MathErr)  
(...)  
// ýalňyslygy isläp geçmek  
  
else {  
    throw;
```

Gaýtalan emele getirme fakty y throw operandynyň ýokdygy bilen bellenýar. Eger kadadan çykma ýok wagtynda gaýtadan emele getirmäge synanyşyk edilen bolsa terminate () funksiýasy çagyrylar.

Gaýtadan emele getirilyän kadadan çykma başlangyç kadadan çykma bolup durýar, ýagny isläp geçijiniň blogy icine alýan binýatlyk klasyň obýekti bilen işlan ýagdaýynda hem döredilen klasyň ilkibaşda emele getirilen obýektini alar.

6. Kadadan çykmalaryň ählisini tutmak

catch işlap geçiji üçin ýagdaý ygylan edilende köp nokat berip bolar, ol islendik ýagdaý bilen birmenzeşligi beryär. Köp nokatly reaksiýa, eger ol bar bolsa, kabir gözegçilikli blogyň reaksiýalarynyň sanawynda iň soňky bolmalydyr.

```
try
{ // Biz zat
edyäris } catch
(...)
{ // Ähli kadadan çykmalary gaýtadan isläp geçmek }
```

7. Isläp geçijileri ýazgy etmegiň tertibi

Döredilen klaslaryň kadadan çykmalarynyň binýatlyk klasyň kadadan çykmalaryny işlap geçijiler tarapyndan tutulyp biljekdigi sebäpli, isläp geçijilerin try instruksiýasynda ýazylan tertibiniň uly ähmiýeti bardyr. Isläp geçijiler olaryň ýazylan tertibinde barlanýar (mysal).

```
try
{ // ... }
(std::ios base::failure) //Giriş/çykyş akymdaky ýalňyslary işlap geçmek

catch (std::exception) // Standart kitaphanalaryň kadadan çykmalary işlap
{ // ... }                       geçmek
catch (...)

// Ähli galan kadadan çykmalary işlap geçmek
```

8. Konstruktorlardaky kadadan çykmalar

Kadadan çykma konstruktordan ýalňyslygy nähili habar bermelidigi meselesini çözmegin usulyny berýar. Konstruktoryň çagyryan funksiýanyň barlap biljek ähmiýetini gaýtarmaýandygy sebäpli, däp bolan (ýagny, kadadan çykmalary isläp geçmegi peýdalanmazdan) alterniwalar sular bolup durýar.

1. Obýekti "dogry däl" ýagdaýda gaýtarmak we peýdalanyjynyň ony barlajakdygyna bil baglamak.

2. Obýekti döretmegiň üstünlikli bolmandygyny görkezmek üçin lokal däl ähmiýeti bermek we peýdalanyjynyň ony barlajakdygyna bil baglamak.

3. Konstruktorla hiç hili inisialyzasiýany amala aşyrmaly däl we obýekt birinji gezek peýdalanylýança peýdalanyjynyň inisializasiýa funksiýasyny (indi ýazylmaly funksiýany) çagyryjakdygyna bil baglamaly.

4. Obýekti inisiallaşdyrylmadyk hökmünde bellemeli we bu obýekt üçin

agza klasyň funksiýasy biriniji gezek çagyrylanda inisiallaşdyrmagy amala aşyrmaly (şunun ýaly funksiýa inisiallaşdyrma üstünlikli bolmadyk ýagdaýynda ýalňyslyk barada habary gaýtaryp biler, ýöne peýdalanyjy funksiýa tarapyndan gaýtarylýan ähmiýeti barybir barlamalydyr).

Kadadan çykma konstruktordan inisiallaşdyrmagyň üstünlikli bolmandygy barada habar berýär. Mysal üçin, Vektor klasy degişli kadadan çykmany emele getirmek bilen, ýadyň örän köp möçberinin talabyndan goranyp biler, (mysal)

```
class Vector

public:
    class Size { ... };
    Vector(int n = 0);
    ...
};
Vector::Vector(int n)
{ if (n < 0 || n > MAX SIZE) throw
Size(); }
```

Wektory döredyän kod indi Vektor::Size ýalňysyny tutup alyp biler we oylanyşykly bir zat etmäge synanyşyp biler.

9. Yalňyslyklar bolup durmayan kadadan çykmalar

Eger kadadan çykma garaşylyan bolsa we programmanyň özüni alyp barşyna yarmaz täsir etmeyän görnüşde tutulan bolsa, onda bu nâme üçin ýalňys hasap edilmeli? Kadadan çykmalary işlap geçmek mehanizmlerine ýene bir dolandyryş gurluşy hökmünde seredip bolar (mysal):

```
: i :, f ;Queue q)
```

```
{ for ( ;; )
    { int x = q. Get ( ) ; // (jet funksiýasy Empty kadadan çykmany
                          Generirileýar(nobat ýok sertinde)

    } catch
    (Queue::Empty) (
return; }
```

Kadadan çykmalary işlap geçmek lokal dolandyryş gurluşlaryna garanyňda truktirleüşi pes mehanizm bolup durýar, onsoňam kadadan çykma hakykatdan hem emele getirilyän halatynda, kadadan çykmalary işlap geçmegin netijesi pes 3I\ar. Sonun üçin hem kadadan çykmalary dăp bolân dolandyryş gurluşlarynyn y (elegant) çözügüt bolup durmayan ýa-da olardan peýdalanmagyň mümkin däl gdaýlarynda peýdalanmak gerek.

Kadadan çykmalary ölçegsiz peýdalanmak düşnüksiz koda getirýär. Adatça ..kadadan çykmalary işlap geçmek ýalňyslary işlap geçmek bolup durýar" diýen nukdaýnazara gollanylmalydyr. Şunun ýaly çemeleşmede kod düşnükli ýagdaýda iki bölege bölünýar: adaty kod we ýalňyslary işlap geçiş kody.

10. Kadadan çykmalaryň spesifikasiýasy

Kadadan çykmalary emele getirmek we tutup almak funksiýalaryň rasyndaky özara hereket etmegiň usulyny iýtgedýär. Sonun üçin hem ygylan etmede funksiýa tarapyndan emele getirilip bilinjek kadadan çykmalaryň toplumyny görkezmek peýdaly bolar (mysal).

```
int f(int n) throw (ex1, ex2);
```

Şunun ýaly ygylan etme '/' funksiýasynyň beýlekileri däl-de, dine ex1, ex2 kadadan çykmalary we bu tiplerden döredilýän kadadan çykmalary emele getirip biljekdigini aňladýar. In möhüm artykmaçlyk funksiýany ygylan etmegiň interfeýse degişli bolup durýandygydyr, ony çagyryanlar görýär.

Kadadan çykmalaryň spesifikasiýasy bolmazdan ygylan edilen funksiýa islendik kadadan çykmany emele getirip bilen diýip hasap edilyär (mysal).

11. Kadadan çykmany emele getiryän Staçs klasy (mysal)

Staçs klasynyň stek dolanda we boş stekden elementi almaga synanyşylanda degişli kadadan çykmalar emele getiriler ýaly görnüşde modifisirleýäris (mysal).

```
#include
<cstdio> class
Stack;
class
StackEmpty {
private:
    Stack * stack; public: StackEmpty(Stack *p) :
    Stack(p) { } Stack* GetPtrO { return Stack; }
}; class
StackFull {
private:
    Stack * stack; int n ; public:
    StackFull(Stack *p, int i) : Stack(p), n(i) {
    } Stack* GetPtr() { return Stack; } int
    GetValue ( ) { return IT; } }; class Stack {
private:
    enum { SIZE = 100 }; int
    Stack[SIZE]; int * cur;
    public: Stack() { cur =
    Stack; } ~Stack() { }
    int Push(int n); int
    Pop();
    int IsEmpty() const { return cur == Stack; }
    int operator >> (int& s) { s = Pop(); return
    s; } int operator << (int s) { return
    Push(s); }
}; int Stack::Push(int
n) { if (cur - Stack <
SIZE)
    { *cur++ = n; return n;
    } else throw
    StackFull(this, n);
} int Stack::Pop() { if
(cur != Stack) return *-
```

```

-cur; else throw
StackEmpty(this); }

void
main() {
    Stack s;
    int n;
    try { s <<
        1;
        s <<
        2; s
        << 3;
        s << 4
        ; s < <
        5;
        s >> n;
        printf("%d\n",
n); s >> n;
        printf("%d\n",
n); s > > n ;
        printf ( "%d\n", n)
        ; s > > n ;
        printf("%d\n", n); } catch (StackFull s) {
printf("Attempt to put a value %d to the full Stack at
the address %p\n", s.GetValue(), s.GetPtr()); } catch
(StackEmpty s) { printf("Attempt to get a value from
the empty Stack at the address %p\n", s.GetPtr()); } }

```

EDEBIÝATLAR

1. Türkmenistanyň Konstitusiyasy. Aşgabat, 2008.
2. Gurbanguly Berdimuhamedow. Ösüşin täze belentliklerine tarap. Saýlanan eserler. I tom. Aşgabat, 2008.
3. Gurbanguly Berdimuhamedow. Ösüşin täze belentliklerine tarap. Saýlanan eserler. II tom. Aşgabat, 2009.
4. Gurbanguly Berdimuhamedow. Garaşsyzlyga guwanmak, Watany, halky söýmek bagtdyr. Aşgabat, 2007.
5. Gurbanguly Berdimuhamedow. Türkmenistan – sagdynlygyň we ruhubelentligiň ýurdy. Aşgabat, 2007.
6. Türkmenistanyň Prezidenti Gurbanguly Berdimuhamedowyň Ministrler Kabinetiniň göçme mejlisinde sözlän sözi. (2009-njy ýylyň 12-nji iýuny). Aşgabat, 2009.
7. Türkmenistanyň Prezidentiniň “Obalaryň, şäherleriň, etrapdaky şäherçeleriň we etrap merkezleriniň ilatynyň durmuş-ýaşayyş şertlerini özgertmek boýunça 2020-nji ýyla çenli döwür üçin” Milli maksatnamasy, Aşgabat, 2007.
8. “Türkmenistany ykdysady, syýasy we medeni taýdan ösdürmegiň 2020-nji ýyla çenli döwür üçin Baş ugry” Milli Maksatnamasy, “Türkmenistan” gazetini, 2003-nji ýylyň 27-nji awgusty.
9. “Türkmenistanyň nebitgaz senagatyny ösdürmegiň 2030-njy ýyla çenli döwür üçin Maksatnamasy”. Aşgabat, 2006.
10. Г.Шилдт. С# 4.0. Полное руководство, Москва-Санкт Петербург, 2011.
11. М.Е.Фленов. Библия С#. 2-ое издание. СанктПетербург, 2011.
12. The C Programming language, Second Edition, Brain W.Kernigan, Deniss M. Ritchie. Colorado Uniwersiteti, 1982.
13. C at Any Speed, Richard S. Wiener, Uniwersity of Colorado at Colorado Spring Colorado Uniwersiteti, 1988.
14. Б. Керниган, Д.Ритчи Язык программирования С++, Москва “Финансы и статистика” – 1992.
15. Р.Уинер. Язык ТУРБО С, Москва, 1991.
16. Дэвид Дж.Круглинский, Скотт Уингоу, Джордж Шеферд (Для профессионалов) Программирование на Microsoft VISUAL С++ 6.0, Санкт-Петербург, Москва, Харьков, Минск, 2002.
17. Н.Культин С/С++ в задачах и примерах, Санкт-Петербург “БХВ-Петербург”, 2008.

MAZMUNY

Giriş	1
Programma ýazmagyň düzgüni	3
C++ diliniň mümkinçiliklerine syn	6
Üýtgeýän ululyk we arifmetiki aňlatma	9
for operatory	15
Atlandyrylan konstantalar(hemişelikler)	16
Literany çapa çykarmak	17
Faýlyň kopiýasyny almak	17
Literi sanamak	19
Setiri sanamak	21
Söz sanamak	22
Massiwler	23
Funksiýalar	26
Argumentler. Bahasy boýunça çagyrmak	29
Literleriň massiwi	30
Daşky üýtgeýän ululyklar we olaryň täsir edýän çägi	33
Tipler, operatorlar we aňlatmalar	37
Üýtgeýän ululyklaryň atlary	37
Maglumatlaryň tipleri we ölçegleri	38
Hemişelikler	39
Deklarirlemek	42
Arifmetiki operatorlar	43
Gatnaşyk operatorlary we logiki operatorlar	43
Tipleri özgertmek	44

Inkrement we dekrement operatorlar	48
Bitleýin operatorlar	50
Funksiýalary döretmek we ulanmak. Funksiýany döretmek	51
Funksiýanyň argumenti	52
Return operator	53
Adresi tapmak, & - operator	53
Funksiýalar, funksiýanyň täsir edýän çägi we ýadyň klaslary	55
Maglumatlaryň şahalaýyn gurluşy	58
C++-de giriş çykyş	63
C++ diliniň ýörite serişdeleri	68
C++ programmirleme diliniň esaslary. Visual programmirleme	74
Microsoft Visual Studio wizual programmirleme dili	84
Maglumatlaryň esasy görnüşleri, olaryň üstünde ýerine ýetirilýän amallar	88
Operatorlar barada düşünje, görkezijiler we salgylar	131
Funksiýalar we görkezijiler bilen işlemek. Rugsat	138
Klaslar we obýektler bilen işlemek. Programmanyň koduna düşündiriş belliklerini goýmak (comments). Gütadan ýüklemek	143
Edebiýatlar	161