МУНДАРИЖА

Кириш

Бугунги кунда илм-фан жадал тараққий этаётган, замонавий ахборот коммуникация воситалари кенг жорий этилган жамиятда барча фан соҳаларида билимларнинг тез янгиланиб бориши, таълим олувчилар олдига уларни тез ва сифатли эгаллаш билан бир қаторда, мунтазам ва мустақил равишда билим излаш вазифасини қўймоқда.

Ҳозирги кунда замонавий компьютер технологиялардан бири бўлиб объектларга мўлжалланган дастурлаш ҳисобланади. Объектларга мўлжалланган дастурлашдан фойдаланиш мураккаб дастурларни уларни технологик хусусиятларини ҳисобга олган ҳолда ишлаб чиқариш имкониятини яратди.

Ахборотлар технологиясининг асосий негизи хозирги замон компьютерлари хисобланиб, уларни фан-техника, ишлаб чикариш ва халк хўжалигининг барча тармокларида кенг кўлланилишига эришилди.

Замонавий дастурлаш таъминотининг мураккаблиги уни яратувчиларидан янада келажаги бор технологияларни билишни талаб килди.

Хозирги вактда Visial Basic га ухшаш CASE ва график курилмалардан канчалик кенг фойдаланилаётган булса хам, лекин дастурларни ёзиш учун албатта дастурчилар керак ва мухим дастурларни С++ дастурлаш тилида ёзиш керак. Агар сизга С++ да фойдали дастурлар керак булса, бунинг учун объектларга мулжалланган дастурлашнинг энг янги тажрибаси керак булади.

Ушбу курс ишида динамик хотирани ташкил этиш дастурини тузиш учун, Си++ дастурлаш тилидан фойдаланган холда, куйилган масалаларни ечиш усуллари билан танишамиз.

1.Динамик хотирани ташкил килиш масаласининг куйилиши

Ушбу курс ишида динамик хотирани ташкил этиш масаласи кўрилади.

Си++ тилида ўзгарувчилар хотира синфига караб статик (static) ва динамик (auto,register)га бўлинади.

Статик ўзгарувчилар программанинг бутун бажарилиш даврида мавжуд бўлади.

Динамик ўзгарувчилар эса, функцияга кириш олдидан ташкил килинади ва функциядан чикишда йўкотилади. Динамик ўзгарувчилар ўз навбатида хотирада жойлашадиган (auto) ёки регистрларда жойлашадиган бўлиши мумкин.

Ушбу курс ишида динамик хотирани ташкил этишнинг мавжуд усулларини тахлил килинади. Назорат мисоллари учун дастур тузилади ва натижалар келтирилади.

2.Назарий кисм

2.1. C++ дастурлаш тилида хотира синфи

Юкорида айтиб утилганидек, Си++ тилида ўзгарувчилар хотира синфига караб статик (static) ва динамик (auto,register)га бўлинади.

Динамик ўзгарувчилар, функцияга кириш олдидан ташкил килинади ва функциядан чикишда йўкотилади. Динамик ўзгарувчилар ўз навбатида хотирада жойлашадиган (auto) ёки регистрларда жойлашадиган бўлиши мумкин.

Хотира синфи типни спецификацияси олдидан ўзгарувчини эълон килаётганимизда аник кўрсатилиши мумкин. Агар хотира синфи кўрсатилмаган бўлса, у холда у ўша ўзгарувчи эълон килинган жойга боглик холда аникланади. Кандайдир функциянинг ичида эълон килинган ўзгарувчилар хотирада жойлашадиган динамик ўзгарувчилардир. Кандайдир функцияни ичида эълон килинган ўзгарувчиларда, хотира синфидан катъий назар, факат шу функциянинг ичида фойдаланиш мумкин. Функциялари билан бир хил даражада эълон килинган ўзгарувчилар, яъни олдида static ёки extem деган ключ сўзи бўлган. ўзгарувчилар программада ихтиёрий ерда фойдаланилиши мумкин, агарда программа бир неча программани файллардан тузилган бўлса хам. Функциялар билан бир хил даражада деб эълон килинган ўзгарувчилар, яъни олдида static ключ сўзи турган ўзгарувчилар хамма функцияларда ишлатилиши мумкин, лекин факат ўзи эълон килинган файл учун.

Си++ тилида ихтиёрий программа бу ташки объектлар тўплами – яъни ўзгарувчилар ёки функциялар тўпламидир. Ташки ўзгарувчилар ихтиёрий функциядан ташкарида аникланган ва кўп функциялар учун фойдаланиши мумкин. функцияларнинг ўзи хар доим ташкидир, чунки тилнинг ўзи функцияни бошка функция ичида аниклашга имкон бермайди.

Кўп алгоритмик тилларда маълумотлар бериладиган параметрлар, массивлар статистик объектлар деб аникланарди, чунки уларга олдиндан канча микдорда кийматлар берилиши маълум хисобланарди. Шу сабаб хар

бир программада маълумотлар хотиранинг кай кисмини эгаллашини аввалданок билиш мумкин эди.

Си тилида динамик объектлар тушунчаси киритилган. Бундай объектлар ўлчови программа ишлаш шароитига караб ўзгаради ва программа динамик объектларга бажарилиш фаолиятида керагича (сўрокка караб) жой ажратади. Мисол учун тузилган программа учун 100, 200 ёки 500 та маълумот киритиш лозим бўлса, мос равишда хотирадан шунча ячейка бандланади.

2.2.C++дастурлаш тилининг мухим имконияти (RTTI) маълумотлари

C++ тилининг энг мухим олдинга сурилган жуда камчилик тушунадиган имкониятларидан бири – бу Run-time туридаги маълумотлар хисобланади. Run-time туридаги маълумотлар –бу шундай тизимни ташкил этадики, улар сизнинг дастурингизга объектларни бажарилиш вактида англаш имконини берадилар. Run-time C++ тилининг махсус оригинал булаги эмасдир. Лекин, унинг тил таркибига кушилиши жуда куп вактлардан бери сабрсизлик билан кутилган эди. Чунки шундай катта булмаган, лекин мухим муаммлар синфи мавжудки, уларнинг ечими бевосита Run-time туридаги маълумотларни киритилиши билан осонлашади. Бугунги кунга келиб, C++ нинг барча асосий компиляторлари Run-time туридаги маълумотларга эгадирлар ва бу спецификация дастурлар тупламини амалга оширишда жуда кенг кулланилмокда.

C++ дастурлаш тилида бажарилиш вакти тури хакидаги маълумот (RTTI) куйидаги иккита калит сузлар оркали кулланилади: typeid ва dunamic_cast. RTTI катта булаги C++ да полиморф синфларга нисбатан кулланилади.

2.3. (RTTI) маълумотлардан фойдаланиш зарурияти

Балким бажарилиш вактлари турлари хакидаги маълумот сиз учун янгиликдир, чунки C, Pascal, BASIC ёки FORTRAN каби нополморф дастурлаш тилларида бундай маълумот йук. Нополиморф тилларда у талаб этилмайди, чунки хар бир объектнинг тури компиляция боскичида аникланади (яъни дастурни ёзилиши жараёнида). Лекин, C++ га ухшаш полиморф тилларда шундай холатлар юзага келиши мумкинки, объектнинг тури компиляция боскичида ноаник булади, объектнинг аник табиати дастурни бажарилиш боскичида аникланиши мумкин. C++ дастурлаш тили база синфига курсаткичлар, виртуал функциялар ва синфлар иерархиясидан фойдаланган холда полиморфизмни амалга оширади. База синфига курсаткичлар бундай йуналишда ушбу база синфига тегишли объектларни ёки ушбу база синфидан келтирилиб чикариладиган ихтиёрий объектга курсатиш учун фойдаланилиши мумкин. Шундай килиб, хар бир аник вакт ичида бундай курсаткич кандай объект турига курсатишини аввалдан аниклаш мумкин эмас. Объектнинг турини аниклаш дастурни бажарилиш вактида амалга оширилиши керак. Купчилик холларда буни амалга ошириш осонгина RTTI сиз амалга ошади, лекин RTTIга зарурият тугиладиган холатлар хам куплаб учраб туради.

2.4.Объект синфига хотиранинг динамик тасимланиши

Объект синфига хотиранинг динамик таксимланишда дастурни бажарилиш боскичида озод (динамик) хотира хисобидан ажратилади. Бу ерда аник хотира ажратиш ва сохаларни озод килиш дастурчи томонидан аникланган вактда амалга оширилади ва у томонидан назорат килиб турилади.

Аммо объектни узига хотирани статик ажратилган вактда хам унинг алохида майдонларига динамик равишда ажратилиши мумкин. Шунинг учун объектни узига ва унинг майдонларига хотирани кандай ажратилганлигига боглик холда улар куйидаги турларга булинадилар:

динамик майдонли статик объектлар;

статик майдонли динамик объектлар;

динамик объектли динамик объектлар.

Узининг ишлатилишига кура юкоридаги объектлар баъзи бир хусусиятларга эгадирлар.

3.Асосий кисм

3.1.Динамик майдонли статик объектлар

C++ дастурлаш тилида ихтиёрий турдаги уша вактда аникланган узгарувчига динамик хотира ажратиш мумкин. Бунинг учун бир канча усуллар мавжуд булиб, улардан энг кулайи new функциясидан фойдаланишдир. new функцияси томонидан яратилган механизмнинг кулайлиги шундаки, бошка холатлардаги каби, унинг кулланиши учун аник улчовли хотира ажратилиши талаб этилмайди. Зарур булган улчов узгарувчи тури оркали автоматик равишда аникланади. New амали узгарувчини жойлашиши учун хотира ажратади ва ажратилган соха манзилини, курсаткични керакли турга айлантириб, кайтаради. Агар кандайдир сабаб туфайли хотира ажратилмай колса, у холда функция NULL курсаткич кайтаради. new функцияси билан ажратилган хотира delete функцияси ёрдамида узгарувчини турига мос равишда озод килинади.

Оддий узгарувчиларга динамик хотира ажратиш жуда хам тугри эмас, чунки хотира факатгина узгарувчининг узигагина эмас, балки хотира ажратилиши хакидаги маълумотга хам сарф этилади. Шу сабабли, хотиранинг динамик таксимланиши мураккаб берилганлар структуралари билан ишлаш вактида максадга мувофикдир. Агар келгуси майдоннинг улчами ноаник булса, у хисоблашлар жараёнида аникланади ёки кирувчи маълумотлардан боглик булади.

Купинча хотирани динамик таксимланишини майдонлар сифатида тупламларни, каторларни, структураларни ва уларнинг комбинацияларидан фойдаланувчисинфларда кулланилади. Бу холда майдон мос турдаги узгарувчига курсаткичга эга булади. Курсаткич узи аникланган берилганлар структурасига хотира ажратилишида уз кийматини олади. Ушбу хотирани озод этилиши узгарувчини турига мос равишда курсаткичдаги манзил буйича амалга оширилади.

Бундай синф конструктори купинча узгарувчиларга зарур буладиган хотира улчовини ажратишни амалга оширади, бу ерда уларниг манзиллари

мос курстакичларга узлаштирилади, шунингдек мумкин булган хотирани мавжудлигини назорат килиб туради.

Синф деструктори эса объектни йукотилишида унга ажратилган хотирани озод килади, чунки бу хотира уз-узидан озод этилмайди.

3.2.Статик майдонли динамик объектлар

Худди бошка турдаги берилганларга ухшаш ихтиёрий турдаги объектга курсаткич оркали мурожаат килиш мумкин. Ушбу холатда кийматни унга «манзилни олиш» амали ёрдамида узлаштириш мумкин (ушбу холда у статик объектни хам манзиллаштира олади). Лекин курсаткични озод соханинг объектига динамик таксимлаш механизмини амалга ошириш учун фойдаланиш фойдалирокдир. Объект тури кандайдир синф булганлиги учун, объектларга кулланганда new функциясига куйидаги мурожаат куриниши кулланади.

< объектга курсаткич исми>= new<синф исми>;

ёки

<объектга курсаткич исми>=new<синф исми> (<параметрлар руйхати>);

Иккинчи куриниш конструкторда параметрлар руйхати булган холда кулланилади.

delete функцияси объект исмигагина курсатишни талаб килади.

Агар динамик объектлар тупламидан фойдаланилаётган булса, у холда унги хотирани куйидагича ажратиш мумкин:

-ёки уни дастурда аникланиш вактида (тупламнинг барча объектларига тенг хажмда) битта узлуксиз булак оркали, масалан:

B mas[] =new B[n]; - ёки тупламнинг хар бир аник элементига алохида циклда, масалан: for(i=0; i<n; i++) mas[i]=new B;

Ажратилган хотирани у кандай ажратилган булса, шундай озод килинади, масалан: delete[] mas; ёки for(i=0; i<n; i++) mas[i]= delete B;

Кандайдир синфнинг объектларининг тупламини аниклашда ушбу синфнинг конструктори чакирилади, бу эса синфни ифодалашда конструкторни ёки инициализация килинмаган конструкторни булишини талаб килади.

Яна шу уринда полиморф синфларнинг динамик объектлари билан ишлаш хусусиятларини таъкидлашимиз зарур. C++ курсаткичларга база синфларини ихтиёрий синфнинг объектлари манзилини узлаштириш

имкониятини беради. Ушбу холатда келтирилган синфда ифодаланган объект майдонларига мурожаат муаммолари юзага келади:

- база синфи объекти курсаткичи унинг майдонлари билан богланган ва унинг келтирилган синфда ифодаланганмайдони у учун куринмасдир. Шунинг учун база синфининг келтирилган синф объекти майдонларига курсаткич оркали мурожаат этишда тил курилмалари оркали курсаткис турини аник кайта аниклаш зарур;

-бундан ташкари, агар база синфига курсаткични аниклашда келтирилган синф динамик объекти ташкил этилса, у холда бундай объектни йукотиш вактида база синфи деструктори чакирилади ва хотира тиник озод этилмайди, чунки деструктор озод этилаётган хотирани улчамларини тугри аниклай олмайди. Ушбу муаммо виртуал дестукторни куллаш оркали хал килинади. Агар база синфи деструкторини ифодалашда уни virtual деструктори билан ифодаласак, у холда келтирилган синф деструкторлари хам виртуал булади. У холда объектни delete оператори ёрдамида база синфига курсаткич йукотиш вактида барча келтириб чикарилган синфларнинг деструкторлари тиник чакирилади. Ушбу барча хусусиятлар полиморф объектлар билан ишлашда уринлидир.

3.3.Динамик майдонли динамик объектлар

Бундай объектлар динамик майдонли статик объектлар ва динамик объектларни характерли хусусиятларини узида акс эттиради. Барча юкорида караб чикилган мисоллар жуда катта ва чегарасиз динамик хотира булишини англатади. Лекин хакикатда бу хотира чекланган, ва жуда катта объектлар сонида ва уларнинг майдонларига ажратилган жуда катта хажмдаги хотирада, шундай пайт келиши мумкинки, качонки хотира ёки объектга, ёки унинг майдонига ажратилмай колиши мумкин. Бундай холларда конструкторда ёки таркатишни бажараётган усулда хотира мавжудлигини текширишни назарга олиш керак. Хотира етишмаган холат юзага келганда хотиранинг нима учун етишмаганлигини аниклаш зарур (объектни ёки майдонни жойлаштириш учун), сунгра мавжуд холатдан акл билан чикиш зарур.

3.4.Динамик хотирани таксимлаш масаласига доир назорат мисоли Динамик майдонли объектларни ташкил этиш ва йукотиш.

```
#include<string.h>
#include<iostream.h>
#include<conio.h>
class Tmass
{public: int *mas; //бутун сонлар тупламига курсаткич
      int size; //туплам улчами
     void print(void);
Tmass() {};
Tmass(int Size);
-Tmass() {delete [] mas; cout<<"Деструктор" <<endl;}
};
Tmass ::Tmass(int Size)
{ cout<<"Конструктор" <<endl;
mas=new int [Size]; //тупламга хотира ажратиш
size=Size;
cout<< "Туплам кийматини " <<size;
cout<<" киритинг: << endl;
for(int i=0; i<size; i++) cin>> mas[i];
}
void Tmass:: print(void)
{
cout<< "туплам киймати: " << endl;
for(int i=0; i<size; i++) cout<<mas[i]<<" ";
cout<<endl;
}
void main()
{
clrscr();
```

```
Tmass aa(6), cc(4); //синф конструктори икки марта чакирилади
aa.print(); cc.print();
getch();} //  иккита объект учун деструктор чакирилади
```

Киритиш-чикариш амалларини кайта аниклашда объект майдонлари аник кийматлари берилганларни киритгандан сунг аник булади. Шунинг сабабли хотиранинг фойдалирок ва тугрирок фойдаланилиши учун, ушбу объект майдонларига синф конструктори ёрдамида эмас, балки аник хотира, яъни киритиш амали вактида ажратилиши максадга мувофикдир.

5.Курс ишини бажаришда фойдаланилган Си++ дастурлаш тили хакида асосий тушунчалар

Ҳозирги кунда замонавий компьютер технологиялардан бири бўлиб объектларга мўлжалланган дастурлаш ҳисобланади. Объектларга мўлжалланган дастурлашдан фойдаланиш мураккаб дастурларни уларни технологик хусусиятларини ҳисобга олган ҳолда ишлаб чиқариш имкониятини яратди.

Дастурлаш амалиёти юзага келган даврдан бошлаб технологик кўринишларни ривожлантира бориб ва улар асосида мураккаб дастур тизимларини яратиш йўли билан дастурлар ишлаб чиқариш жараёнларини осонлаштирадиган дастурлаш қурилмаларининг яратилишини талаб қилиб келди.

Си++ -бу ифодаларнинг ёзилишининг қулай усули, берилганлар структурасининг замонавий бошқарув механизми ва бой операторлар тўпламига эга бўлган дастурлашнинг универсал тилидир. Си++ дастурлаш тили „жуда ҳам юқори даража” тили ҳам , „катта„ тил ҳам эмас, бирон бир аниқ соҳага фойдаланишга махсуслаштирилган ҳам эмас. Лекин унга қўйилган чегараларнинг йўқлиги ва универсаллиги бу дастурлаш тилининг баъзи кучли тилларга кўра кўпгина масалаларни қулай ва фойдалироқ ечилишига имкон беради.

Тажрибаларимиздан келиб чиққан ҳолда шуни айтишимиз мумкинки, Си++ дастурлаш тили –қулай, ифодали ва кенг масалалар синфи учун керак бўладиган дастурлаш тилидир.

Си++ тилида ўзгарувчилар хотира синфига қараб статик (static) ва динамик (auto,register)га бўлинади.

Статик ўзгарувчилар дастурнинг бутун бажарилиш даврида мавжуд бўлади.

Динамик ўзгарувчилар эса, функцияга кириш олдидан ташкил қилинади ва функциядан чиқишда йўқотилади.

Хотира синфи типни спецификацияси олдидан ўзгарувчини эълон килаётганимизда аник кўрсатилиши мумкин. Агар хотира синфи кўрсатилмаган бўлса, у холда у ўша ўзгарувчи эълон килинган жойга боглик холда аникланади. Кандайдир функциянинг ичида эълон килинган ўзгарувчилар хотирада жойлашадиган динамик ўзгарувчилардир. Хеч кандай функцияни ичида жойлашмаган, функциялар билан бир хилда эълон килинган ўзгарувчилар статик ўзгарувчилар дейилади. Кандайдир функцияни ичида эълон килинган ўзгарувчиларда, хотира синфидан катъий назар, факат шу функциянинг ичида фойдаланиш мумкин. Функциялари билан бир хил даражада эълон килинган ўзгарувчилар, яъни олдида static ёки extem деган ключ сўзи бўлган. ўзгарувчилар программада ихтиёрий ерда фойдаланилиши мумкин, агарда программа бир неча программани файллардан тузилган бўлса хам. Функциялар билан бир хил даражада деб эълон килинган ўзгарувчилар, яъни олдида static ключ сўзи турган ўзгарувчилар хамма функцияларда ишлатилиши мумкин, лекин факат ўзи эълон килинган файл учун.

Си++ тилида ихтиёрий программа бу ташки объектлар тўплами – яъни ўзгарувчилар ёки функциялар тўпламидир. Ташки ўзгарувчилар ихтиёрий функциядан ташкарида аникланган ва кўп функциялар учун фойдаланиши мумкин. функцияларнинг ўзи хар доим ташкидир, чунки тилнинг ўзи функцияни бошка функция ичида аниклашга имкон бермайди.

Кўп алгоритмик тилларда маълумотлар бериладиган параметрлар, массивлар статистик объектлар деб аникланарди, чунки уларга олдиндан канча микдорда кийматлар берилиши маълум хисобланарди. Шу сабаб хар бир программада маълумотлар хотиранинг кай кисмини эгаллашини аввалданок билиш мумкин эди.

Си++ тилида динамик объектлар тушунчаси киритилган. Бундай объектлар ўлчови программа ишлаш шароитига караб ўзгаради ва программа динамик объектларга бажарилиш фаолиятида керагича (сўрокка караб) жой ажратади.

5.Хулоса

Ушбу курс ишидан олинган асосий натижа ва хулосалар қуйидагича:

Кандайдир синф объектларини ташкил этиш ва йукотишда куйидаги коидаларга амал килинади:

- глобал ва локал статик объектлар main функцияси чакирилгунга кадар ташкил этилади ва дастур тугаши билан йукотилади;

-автоматик объектлар хар гал уларни эълон килинганда ташкил этиладилар ва улар пайдо булган функциядан чикишлари билан йукотиладилар;

- динамик хотира ажратиладиган объект new функцияси билан ташкил этилади ва delete функцияси билан йукотилади.

Натижалар:

1) Динамик хотирани ташкил килиш масаласининг таҳлили келтирилган.
2) Таҳлил қилинган имкониятлар бўйича назорат мисолларининг натижалари келтирилган.
3) Курс ишини бажаришда фойдаланилган объектларга мўлжалланган дастурлаш тили C++ ҳақида асосий маълумотлар берилган.
4) Тахлил килинган мисоллар буйича алгоритм тузилди, дастур матни ёзилди ва назорат мисолларидан фойдаланган ҳолда натижалар олинган.

Фойдаланилган адабиётлар

1.Иванова Г.С., Ничушкина Т.Н., Пугачев Е.К. «Объектно-ориентированное программирование» Изд. МГТУ им.Н.Э.Баумана, 2003 г.

2.Каримов И.А., Жахон молиявий –иктисодий инкирози, Узбекистон шароитида уни бартараф этишнинг йуллари ва чоралари. Тошкент: Узбекистон нашриёти, 2009.

3.Каримов Р,Ирискулов С. «Дастурлаш», Тошкент, «Узбекистон», 2003.

4.Бек Л. «Введение в системное программирование», Москва, «Мир» , 1988 г. «Наука» 1988г.

Иловалар

Дастур 1

```
// Dinamik maydonli ob`ektlarni tashkiletish va yo`qotish
#include<string.h>
#include<iostream.h>
#include<conio.h>
class Tmass
{public: int *mas; //бутун сонлар тупламига курсаткич
      int size; //туплам улчами
     void print(void);
Tmass() {};
Tmass(int Size);
-Tmass() {delete [] mas; cout<<"Деструктор" <<endl;}
};
Tmass ::Tmass(int Size)
{ cout<<"Конструктор" <<endl;
mas=new int [Size]; //тупламга хотира ажратиш
size=Size;
cout<< "Туплам кийматини " <<size;
cout<<" киритинг: << endl;
for(int i=0; i<size; i++) cin>> mas[i];
}
void Tmass:: print(void)
{cout<< "туплам киймати: " << endl;
for(int i=0; i<size; i++) cout<<mas[i]<<" ";
cout<<endl; }
void main()
{clrscr();
Tmass aa(6), cc(4); //синф конструктори икки марта чакирилади
```

```
aa.print(); cc.print();
getch();} //  иккита объект учун деструктор чакирилади
```

Дастур 2

```
// Статик майдонли динамик объектлардан фойдаланиш
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
class Tvector
{ private: int x,y,z;
public:
Tvector() { cout<<"inisalzasiya qilmaydigan konstruktor"<<endl;}
Tvector(int ax, int ay, int az)
{cout<<"Konstruktor"<<endl; x=ax; y=ay; z=az;}
~Tvector(){ cout<<"Destruktor"<<endl;}
void PrintVec();
};
void Tvector::PrintVec()
{ cout<<"Vectorning qiymati: "<<setw(5)<<x<<",";
  cout<<setw(5)<<y<<","<<setw(5)<<z<<"\n";}
int main()
{ Tvector *a, *b;  // Sinf ob'ektlariga ikkita ko`rsatkich aniqlaymiz
//Sinfning dinamik ob`ektlariga hotira ajratamiz
a=new Tvector(12,34,23);  //konstruktor chaqiriladi
b=new Tvector(10,45,56);  //Konstruktor chaqiriladi
a->PrintVec();  // 12,34,23 chiqadi
b->PrintVec();  //10,45,56 chiqadi
//Sinfning dinamik ob`ektlariga ajratilgan hotirani ozod qilamiz
```

```
delete a;   //destruktor chaqiriladi
delete b;   //destruktor chaqiriladi
getch();
}
```