# Оюутан Өгөгдсөн Даалгаварыг Гүйцэтгэх Лабораторийн Ажил (Лаборатори №5)

Т. Оюунболд , Э.Хажидмаа

ХШУИС, МКУТ-ийн оюутан , 15B1SEAS1192@stud.num.edu.mn

ХШУИС, МКУТ-ийн оюутан , 15B1SEAS2093@stud.num.edu.mn

# 1. ОРШИЛ/УДИРТГАЛ

Лаборатор № 4 дээр үзсэн сэдвээ цааш үргэлжлүүлэн гүнзгийрүүлэн судалж, асуултанд хариулан, туршиж үзэн тайлбар хийнэ.

2. ЗОРИЛГО

Асуултанд товч бөгөөд оновчтой хариулах Лаборатори №4 - д тодорхойлсон классын нэр:char[20], албан тушаал:char[10] гэсэн гишүүн өгөгдлийг хаяган хувьсагч болгон өөрчилж гараас өгсөн тэмдэгтийн цуваатай яг ижил урттай санах ой new оператор ашиглан нөөцөлдөг болгох. Үүний тулд анхдагч болон параметертэй байгуулагч функцууд тодорхойлж гишүүн өгөгдөлд гарааны утга онооно. Мөн объект устах үед дээрх хоёр гишүүн өгөгдөлд нөөцөлсөн санайх ойг чөлөөлдөг болгож өөрчилнө.

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц( Constructor )

Объектийн гишүүн өгөгдлүүдэд гарааны утга оноох, санах ой нөөцлөх үүрэгтэй тусгай зориулалтын функцийг байгуулагч функц гэнэ.

employee(){

strcpy(name, “”);

basicpay = 0;

allowance = 0;

}

3.1.1 Байгуулагч функцийн шинжүүд

- Байгуулагч бол public функц юм.

- Байгуулагч нь дотоод мөр функц байж болно.

- Байгуулагч функц объект байгуулах үед автоматаар дуудагдах ба объект байгуулсны дараа дуудахыг хориглодог. ( дуудаж болохгүй)

- Байгуулагч функцийн нэр классын нэртэй яг ижил байна.

- Байгуулагч функц утга буцаахгүй. (void биш)

- Байгуулагч функцийг дахин тодорхойлж болно. Класс нь олон байгуулагчтай байж болно.

- Объект хэрхэн байгуулагдахыг байгуулагч функцээр тодорхойлно.

- Байгуулагч функц өөрөө төрөл хувиргалт хийх аргатай.

- Байгуулагч функцийг аргумент нь объект байвал түүнийг хувилна.

- Хаягаар нь байгуулагч функц руу хандаж болдоггүй.

- Байгуулагч функц нь new ба delete операторыг ил бишээр дуудаж хэрэглэнэ.

- Байгуулагч нь бусад гишүүн функцийн адил удамшдаггүй.

3.2 Анхдагч байгуулагч (default construction)

Параметргүй байгуулагчийг анхдагч байгуулагч гэнэ. Бүх объектийг адилхан байгуулах үед хэрэглэнэ. Хэрэв класс ямар нэгэн байгуулагч функц тодорхойлоогүй байвал компайлер анхдагч байгуулагчийг хоосноор тодорхойлж өгдөг.

3.3 Функц дахин тодорхойлох

Нэр нь ижил хэдий ч параметрийн тоо, төрөл, дараалал нь өөр функцуудийг дахин тодорхойлогдсон функц гэж нэрлэнэ.

Функцийн нэр = ялгац ( identifier )

Функцийн зарлалт = бичдэс (signature)

Дахин тодорхойлж байгаа функцуудын бичдэс нь өөр байх ёстой.

3.4 Параметртэй байгуулагч функц

Функц нь параметртэй байна. Жишээ нь:

employee (char n[], int a, int b);

3.4.1 Параметртэй байгуулагчийн давуу талууд

- Байгуулагч функцийн тодорхойлолт дотор гишүүн өгөгдөлд утга оноохыг шаарддаггүй.

- Объект байгуулахдаа дамжуулан өгөх утгыг гаднаас авна.

- Объектоор дамжуулж авсан өгөгдлийг автоматаар гишүүн өгөгдөлд хадгална.

- Бусдаас өөр утгатай объект байгуулахыг зөвшөөрнө.

- Гишүүн өгөгдөлд гарааны утга оноож болно.

- Анхдагч байгуулагчийн нэгэн адилаар байгуулагдах объектод ой бэлдэнэ.

3.5 Анхны утгатай параметр бүхий функц

Объектын гарааны утгыг аргумент хэлбэрээр параметртэй байгуулагч руу дамжуулж болдог. Ийм гарааны утгыг байгуулагдах объектоос дамжуулж өгнө. Жишээ нь:

empolyee:: employee (char n[], int a, int b){

strcpy (name, n);

basicpay=a;

allowance=b;

}

3.6 Гарааны утгатай байгуулагч

Параметртэй байгуулагчаар объект байгуулах бүрдээ гаднаас бүх аргументээр нь байнга хангаж байх ёстой. Ийм байгуулагчийн зарим аргументийг гаднаас өгөх, заримыг нь анхнаасаа зааж өгөх тогтмол утгаар нь хэрэглэхийг програм шаарддаг. Энэ нь анхдагч байгуулагч ба параметртэй байгуулагч хоёрын аль алинд нь давуу чанарыг өөртөө шингээсэн байгуулагч хэрэгтэй гэсэн үг юм. Ийм байгуулагч бол гарааны утгатай байх юм. Жишээ нь:

employee(char n[], int a, int b= 2000){

strcpy (name, n);

basicpay = a;

allowance = b;

}

3.7 Устгагч функц

Объект байгуулагдахад түүнд зориулан ой бэлдэхээр анхдагч эсвэл хэрэглэгчийн тодорхойлсон байгуулагч функц үргэлж дуудагдана. Тэгэхлээр, ямар нэгэн байгуулагч функц дуудагдахгүйгээр объект байгуулагдахгүй. Үүнтэй адилаар, объект нь устгагч (destructor) функцгүйгээр устаж чадахгүй. Устгагч функц нь цаашдаа хэрэглэгдэхгүй объект устахад түүний эзэмшиж байсан ойг чөлөөлнө. Хэрэв класс дотор устгагч функцийг илээр тодорхойлж өгөөгүй тохиолдолд объект устахад анхдагч (default) устгагч функцийг С++ компайлер нэмж тодорхойлж хэрэглэнэ.

3.8 Устгагч функцийн тодорхойлолт

Устгагч функц нь классынхаа гишүүн функц болно.Функцийн нэр нь классынхаа нэртэй ижил байна. Зөвхөн өмнө ~ тэмдэгтэй бичигдэнэ. Жишээ нь:

~employee();

3.9 Устгагч функцийн шинжүүд

- Устгагч функц гол төлөв public шинжтэй.

- Устгагч функц нь дотоод мөр функц байж болно.

- Класс нь ганц устгагч функцтэй байна.

- Устгагч функц нь const, static ба volatile төрлийнх байж болохгүй.

- Устгагч функц нь virtual байж болно.

- Объект устах үед устгагч функцийн зөвхөн С++ компайлер дуудна.

- Устгагч функц утга буцаахгүй.

- Устгагч функц параметр авахгүй.

- Объект устахад түүний эзэмшиж байсан ойг устгагч функц чөлөөлнө.

- Устгагч функц нь байгуулагч функцийн адилаар нэмэлт үйлдэл хийж болно.

- Устгагч функц рүү хаягаар хандаж болдоггүй.

- Классын бусад адилаар удамшдаггүй.

3.10 Устгагч функцийн хэрэглээ

Объектод зориулан байгуулагч функцийн бэлдэх ойг хэрэглэж дууссаны дараагаар чөлөөлөхөд устгагч функцийг хэрэглэнэ. Класс олон байгуулагчтай байж болох ч зөвхөн ганц устгагч функцтэй байна. Устгагч функцийг класс дотор зарлаж классын гадна талд тодорхойлж болдог.

3.11 Объектон хүснэгт ба байгуулагч функц

employee e[3];

Энэ командаар employee төрлийн 3 объекттой е хүснэгт байгуудагдана. Ямар нэгэн байгуулагч функц классын тодорхойлолтод байгаа бол түүнийг програм үргэлж дуудаж хэрэглэнэ. Иймээс классын тодорхойлолт нь ямар ч байгуулагчгүй эсвэл ядаж анхдагч байгуулагчтай байхыг програм шаарддаг. Дээрх хүснэгтийн тухайлсан бүтвэрт утга олгохдоо байгуулагч функцийг доор үзүүлснээр дуудаж болно.

e[3] = employee (“Bat”, 5000, 2000);

Харин параметртэй байгуулагч функцийг хэрэглэн объектон хүснэгтийн бүтвэрт гарааны утга оноож болох ба ингэхдээ байгуулагч функцийг бүтвэр бүрд хэрэглэнэ.

employee e[3] = { employee (“1”, 1, 1 ),

employee (“2”, 2, 2 ),

employee (“3”, 3, 3 )};

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Олон ажилчин бүртгэж ажилчдыг нэрээр нь эрэмбэлэх

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

{

for(j = i+1; j < n; j++)

//omnoh ajiltnii ner daraagiin ajiltnii nernees omno n bwal swap hiine

if(strcmp(emp[i].getName(),emp[j].getName()) == 1){

//swap hiih vildel

empk = emp[i];

emp[i] = emp[j];

emp[j] = empk;

}

}

4.2 Шинэ ажилчин бүртгэхдээ тухайн ажилчны ID өмнө бүртгэлтэй эсэхийг шалгаж бүртгэлтэй бол бүртгэхгүй. Өөрөөр хэлбэл ID дахин давхцахгүй болго.

//obiekten hvsneg vvsgene, mun engiin empk nertei obiekt vvsgene

Employee emp[10],empk;

//ashiglah parametervvdee zarlana

int n, k, i, j;

int number, jobTime, count = 0;

char name[10], position[10];

//n ajiltnii toog garaas unshina

cout << "Oruulah ajiltanii too:";

cin >> n;

//0 -ees n hvrtel ajiltanii medeelel garaas unshina

for(k = 0; k < n; k++){

//ahiltanii dugaariig shalgaj utga onooh dawtalt

do{

//shineer oruulah dugaariig garaas unshih

cout << "\t" <<"Ajiltnii dugaar : ";

cin >> number;

//oruulsan dugaariig omnoh obiektiin ajilchdiin dugaartai dawhtsaj bgaa esehiig shalgah dawtalt

for(int i = 0; i < k; i++){

if(emp[i].getNum() == number)

//herew ajilchdiin dugaar dawhtsal tooluuriig nemegdvvlen dawtaltiig duusgana.

count++;

break;

}

//herew tooluur oorchlogdoogvi bol k dahi ajilchnii dugaariig gishvvn ogogdold onoono

if(count == 0){

emp[k].setNum(number);

}

//esregtohioldold aldaag hewlen count iin utgiig 0 bolgoj dahin deerh vildlvvdiig dawtana.

else{

cout << "Ajiltnii dugaar dawhtsaj bna\n Dahin oruulna uu!!\n";

count = 0;

}

//herew k ajilt gishvvn ugugdul anhnii utgaasaa oor garaas utga awhad dawtalt duusna

while(emp[k].getNum() == 0){

cout << "\t" <<"Ajiltnii ner : ";

cin >> name;

emp[k].setName(name);

}

cout << "\t" <<"Alban tushaal : ";

cin >> position;

emp[k].setPos(position);

cout << "\t" <<"Ajilsan tsag : ";

cin >> jobTime;

emp[k].setTime(jobTime);

emp[k].calSalary();

}

# 5. ДҮГНЭЛТ

# Энэ лабораторын ажлаар Лаборатор №4 дээр хийсэн ажлаа цааш үргэлжлүүлэн классын нэр:char[20], албан тушаал:char[10] гэсэн гишүүн өгөгдлийг хаяган хувьсагч болгон өөрчилж гараас өгсөн тэмдэгтийн цуваатай яг ижил урттай санах ой new оператор ашиглан нөөцөлдөг болгосон. Үүний тулд анхдагч болон параметертэй байгуулагч функцууд тодорхойлж гишүүн өгөгдөлд гарааны утга оноож, мөн объект устах үед гишүүн өгөгдөлд нөөцөлсөн санайх ойг чөлөөлдөг болгож өөрчиллөө. Олон ажилчин бүртгэж ажилчдыг нэрээр нь эрэмбэлэж, шинэ ажилчин бүртгэхдээ тухайн ажилчны ID өмнө нь бүртгэлтэй эсэхийг шалгаж бүртгэлтэй бол бүртгэхгүй. Өөрөөр хэлбэл ID дахин давхцахгүй болгох зэрэг өөрчлөлтүүдийг оруулж амжилттай гүйцэтгэлээ.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.

# 7. ХАВСРАЛТ

1. #include <iostream>

#include <string.h>

using namespace std;

class Employee

{

private:

int number;

char \*name;

char \*position;

float salary;

float jobtime;

float bossSalary();

public:

Employee();

Employee(char \*, char \*, int ,float , float);

~Employee();

void setName(char \*);

char \* getName();

void setPos(char \*);

char \* getPos();

void setNum(int );

int getNum();

void setTime(int );

int getTime();

// Garaas ugugdul unshih gishvvn punkts

void showdata();

// salary g/ogogdliin utgiig butsaah gishvvn punkts

float getpay();

// ajilchdiin tsalin tootsooloh g/punkts

float calSalary();

// nemelt tsag tootsoh g/punkts

int addJob(float);

};

Employee::Employee(){

name = new char[7];

strcpy(name," ");

position = new char[10];

strcpy(position,"position");

number = 0;

salary = 0;

jobtime = 0;

}

Employee::Employee(char \*n, char \*p, int a, float s, float j){

name = new char [ strlen(n) + 1 ];

strcpy (name,n);

position = new char [strlen(p) + 1];

strcpy (position,p);

number = a;

salary = s;

jobtime = j;

}

Employee::~Employee(){

delete position;

delete name;

}

void Employee::setName(char \*n){

if(name)

delete name;

name = new char[strlen(n)+1];

strcpy(name, n);

}

char \* Employee::getName(){

return name;

}

void Employee::setPos(char \*p){

if(position)

delete position;

position = new char[strlen(p)+1];

strcpy(position, p);

}

char \* Employee::getPos(){

return position;

}

void Employee::setNum(int n){

number = n;

}

int Employee::getNum(){

return number;

}

void Employee::setTime(int j){

jobtime = j;

}

int Employee::getTime(){

return jobtime;

}

//ajilchnii medeelel haruulah gishvvn punkts todorhoilson heseg

void Employee::showdata(){

cout << "Number = " << number << "\n";

cout << "Name = " << name << "\n";

cout << "Position = " << position << "\n";

cout << "Salary = " << salary << endl;

}

//tsalin tootsoh gishvvn punkts

float Employee::calSalary(){

//garaas unshsan position(alban tushaal) boss hvsnegtiinhtiin utgatai adil esehiig shalgana

if(strcmp(position,"zahiral") == 0)

//adil bwal zahirliin tsalin bodoh g.punktsiig duudna

bossSalary();

//vgvi bol hewiin ajilchnii tsalin bodoj salary-d utgiig onoono

else

//niit awah tsalin / ajillah niit tsag \* (ajilsan honog\* ajilsan tsag)

salary = 300000 / 168 \* (21\*jobtime);

}

//zahirliin tsalin bodoh gishvvn punkts

float Employee::bossSalary(){

//niit awah tsalin / ajillah niit tsag \* (ajilsan honog\* ajilsan tsag)

salary = 500000 / 168 \* (21\*jobtime);

}

//nemelt tsag tootsoh g.punkts

int Employee::addJob(float j){

// odort ajilsan tsag ni 0 oos ih 24 oos baga bwal 1 iig butsaana esreg tohioldold 0 iig butsaana

if(jobtime > 0 && jobtime < 24){

jobtime += j;

return 1;

}

else

return 0;

}

//ajilchnii tsalind handah g.punkts

float Employee::getpay(){

//punkts duudsan ved tuhain ajilchnii salary-g butsaana

return salary;

}

int main()

{

//obiekten hvsneg vvsgene, mun engiin empk nertei obiekt vvsgene

Employee emp[10],empk;

//ashiglah parametervvdee zarlana

int n, k, i, j;

int number, jobTime, count = 0;

char name[10], position[10];

//n ajiltnii toog garaas unshina

cout << "Oruulah ajiltanii too:";

cin >> n;

//0 -ees n hvrtel ajiltanii medeelel garaas unshina

for(k = 0; k < n; k++){

//ahiltanii dugaariig shalgaj utga onooh dawtalt

do{

//shineer oruulah dugaariig garaas unshih

cout << "\t" <<"Ajiltnii dugaar : ";

cin >> number;

//oruulsan dugaariig omnoh obiektiin ajilchdiin dugaartai dawhtsaj bgaa esehiig shalgah dawtalt

for(int i = 0; i < k; i++){

if(emp[i].getNum() == number)

//herew ajilchdiin dugaar dawhtsal tooluuriig nemegdvvlen dawtaltiig duusgana.

count++;

break;

}

//herew tooluur oorchlogdoogvi bol k dahi ajilchnii dugaariig gishvvn ogogdold onoono

if(count == 0){

emp[k].setNum(number);

}

//esregtohioldold aldaag hewlen count iin utgiig 0 bolgoj dahin deerh vildlvvdiig dawtana.

else{

cout << "Ajiltnii dugaar dawhtsaj bna\n Dahin oruulna uu!!\n";

count = 0;

}

//herew k ajilt gishvvn ugugdul anhnii utgaasaa oor garaas utga awhad dawtalt duusna

}while(emp[k].getNum() == 0);

cout << "\t" <<"Ajiltnii ner : ";

cin >> name;

emp[k].setName(name);

cout << "\t" <<"Alban tushaal : ";

cin >> position;

emp[k].setPos(position);

cout << "\t" <<"Ajilsan tsag : ";

cin >> jobTime;

emp[k].setTime(jobTime);

emp[k].calSalary();

}

//bubble sort

for(i = 0; i < n; i++){

for(j = i+1; j < n; j++)

//omnoh ajiltnii ner ni daraagiin ajiltnii nerees omno bwal swap hiine

if(strcmp(emp[i].getName(),emp[j].getName()) == 1){

//swap hiih vildel

empk = emp[i];

emp[i] = emp[j];

emp[j] = empk;

}

}

//print jagsaalt

cout << "Jagsaalt:" << endl;

//0 ees n hvrtel eremblegdsen ajilnii medeelliig hewlene

for(k = 0; k < n; k++){

//emp[k] dugaar ajiltnii medeelliig showdata g.punktseer hewlene

emp[k].showdata();

}

return 0;

}