# Классын хүснэгт болон бусад функцийн талаарх ойлголт

# (Лаборатори №4)

Г. Саруул (20B1NUM2095)

ХШУИС, МКУТ-ийн Компьютерийн ухаан хөтөлбөрийн оюутан

# 1. ОРШИЛ

Байгуулагч функц, устгагч функц, дахин тодорхойлогдсон функц болон динамик санах ойн new, delete операторуудын талаар судалснаа хэрэгжүүлсэн.

# 2.ЗОРИЛГО

Өмнөх лабораторийн ажлын хүрээнд тодорхойлсон ажилчин нэртэй классыг илүү сайжруулж ажилчдыг цалингаар нь эрэмбэлнэ.

Үүний тулд:

1. Анхдагч байгуулагч функц

2. Устгагч функц

3. Функц дахин тодорхойлох

4. Динамик санах ой талаар судлах болно.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Байгуулагч функц гэж юу вэ? Анхдагч байгуулагч гэж юу вэ?

Байгуулагч функц нь объектод анхны утга олгох, санах ой бэлдэх үүрэгтэй, объект үүсэх үед хамгийн эхэнд дуудагдан ажилладаг тусгай функц бөгөөд анхдагч байгуулагч функц нь ямар нэг параметр агуулаагүй зөвхөн нас гэдэг гишүүн өгөгдөл байлаа гэж үзвэл гарааны утгыг нь 0 гэж оноох бөгөөд үүнтэй адил өөр үйлдлүүдийг хийж болно. Хэрвээ байгуулагч функцыг тодорхойлж өгөөгүй бол compiler авмоматаар анхдагч байгуулагч функцыг тодорхойлдог. Байгуулагч функцийг тодорхойлохдоо классын нэртэй адил тодорхойлдог. Мөн энэ ямар нэгэн утга буцаадаггүй.

3.2 Устгагч функц гэж юу вэ ?

Устгагч функц нь объектын динамик гишүүн өгөгдлийг санах ойгоос чөлөөлөх функц юм. New оператор ашиглан нөөцөлсөн хувьсагчийг delete оператор ашиглан чөлөөлөх үүрэгтэй функц бөгөөд класс өөрөө цор ганц устгагч функцтэй байх ба энэ функц объект ашиглагдаж дуусаны дараа ажиллана.

3.3 Функц дахин тодорхойлох гэж юу вэ?

Дахин тодорхойлогдсон функц нь өмнө нь тодорхойлогдсон функцтэй адил нэртэй байх ба буцаах утгын төрөл, параметрээрээ ялгаатай байдаг. Хэрэв үгүй бол хооронд нь ялгах боломжгүй болох ба ажиллахгүй юм.

3.4 Динамик санах ой (new, delete оператор ашигласан) болон байгуулагч, устагч функцийг хэрхэн хамтад нь ашиглах вэ?

Динамик санах ойг ашиглахдаа байгуулагч функц дотроо new операторыг ашиглан

объектыг үүсэх үед санах ойгоо нөөцлөөд тухайн объектын санах ойг ашиглан нөөцөлсөн

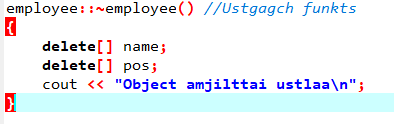
хувьсагчаа ашиглаж дуусмагц автоматаар устгагч функц дуудагдан delete оператороор санах ойгоо чөлөөлнө.

# 4. ХЭРЭГЖҮҮЛЭЛТ

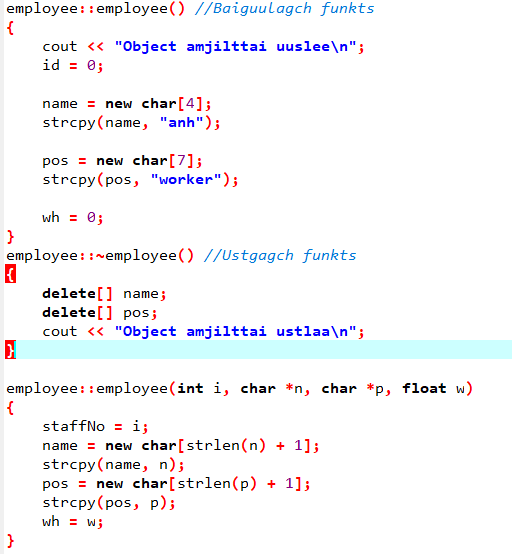
4.1 Lab03-д тодорхойлсон класст анхдагч болон параметртэй байгуулагч нэмж тодорхойлох ба тодорхойлохдоо байгуулагч функцийн үүргийг сайтар хэрэгжүүлсэн байх шаардлагатай.

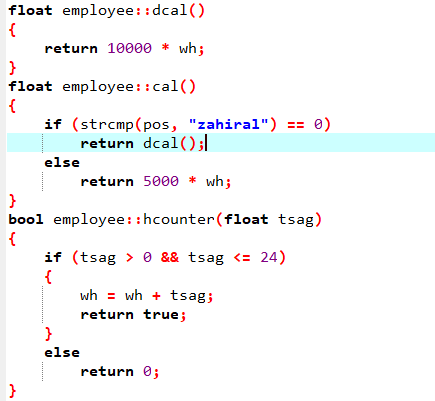
* 1. Lab03-д тодорхойлсон класст устгагч функц тодорхойлж устгагч функц хэзээ дуудагдаж байгааг туршилтад үндэслэн хариул.

Устгагч функц нь объект устгагдах үед дуудагдаж, объект ашиглагдаж дууссаны дараа устгагч функц ажиллаж “Object amjilttai ustlaa” гэж хэвэлнэ.

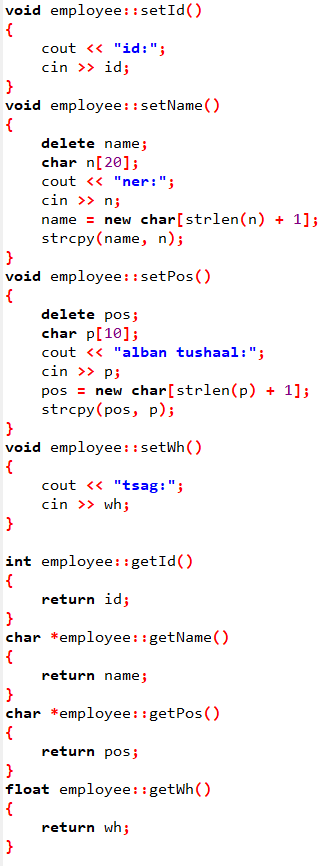


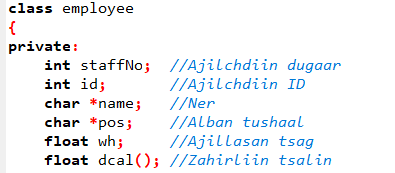
* 1. Lab03-д тодорхойлсон классын захирлын цалин бодох функцийг private хандалтын түвшинтэй болгож өөрчил. Дараа нь цалин бодох функц дотор албан тушаал нь захирал байвал захирлын цалин бодох функцийг цалин бодох функц дотор дуудаж захирлын цалинг бодно. Энд гишүүн функц дотроос гишүүн функц дуудах үйлдлийг хийнэ.



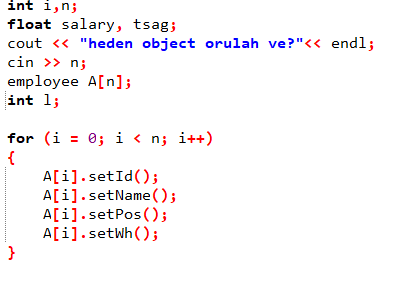


* 1. Мөн гишүүн өгөгдөл бүрийг private хандалтын түвшинтэй болгож лекц дээр заасан set болон get функц бичиж утга оноож, утгыг буцааж авна.



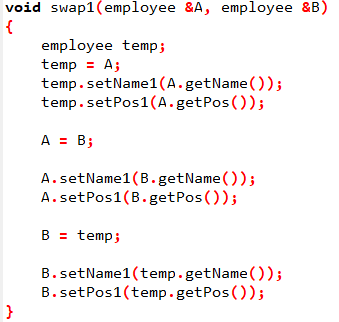


4.5 Ажилчин классын хүснэгт үүсгээд гараас хэд хэдэн ажилчны утга онооно

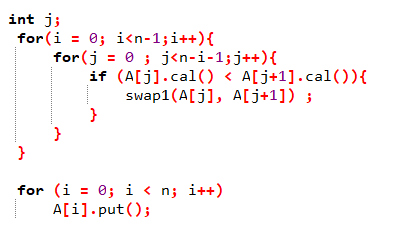


4.6 Утга оноосон хүснэгтийг цалингаар нь эрэмбэлнэ.

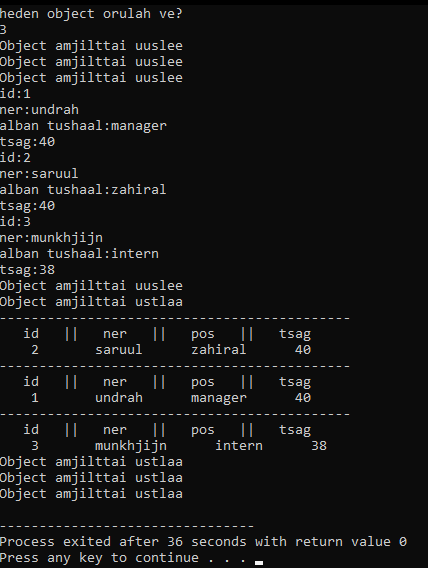
Утга оноосон хүснэгтийг цалингаар нь эрэмбэлэхдээ swap1 гэдэг заалтан хувьсагч параметртаа авч хоёр хувьсагчийг swap хийдэг гишүүн функц тодорхойлсон.



Утга оноосон хүснэгтийг цалингаар нь эрэмбэлсэн нь:



Кодыг ажлуулсан үр дүн:



# 5. ДҮГНЭЛТ

Байгуулагч болон устгагч функцүүд нь классын маш чухал ойлголтууд бөгөөд үүнийг зөв зохион байгуулснаар код хурдан алдаагүй ажиллах боломжтой юм. Харин эсэргээрээ санах ойг динамикаар нөөцөлсөн байж байгаад, устгагч функцээ буруу тодорхойлсон эсвэл тодорхойлоогүй тохиолдол санах ой хий дэмий зай эзлэх, алдаа гарах зэргээс үүдээд программ удааширч гацахад хүрдэг.

# 6. АШИГЛАСАН МАТЕРИАЛ

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

2. <http://www.cplusplus.com/doc/tutorial/>

# 7. ХАВСРАЛТ

#include <iostream>

#include <string.h>

using namespace std;

class employee

{

private:

int staffNo; //Ajilchdiin dugaar

int id; //Ajilchdiin ID

char \*name; //Ner

char \*pos; //Alban tushaal

float wh; //Ajillasan tsag

float dcal(); //Zahirliin tsalin

public:

employee(); //Baiguulagch funkts

~employee(); //Ustgagch funkts

employee(int i, char \*n, char \*p, float w); //Parametrt baiguulagch funkts

void setId(); //Garaas id avah, ugugdul uguh funkts

void setName(); //Garaas neriig avah, ugugdul uguh funkts

void setName1(char\* str);

void setPos(); //Garaas alban tushaaliig avah, ugugdul uguh funkts

void setPos1(char\* str);

void setWh(); //Garaas tsagiig avah, ugugdul uguh funkts

int getId(); //ID butsaah.

char \*getName(); //Ner butsaana.

char \*getPos(); //Alban tushaal butsaana.

float getWh(); //Tsag butsaana

void put(); //Objectiin gishuun ogogdliig hevlene.

float cal(); //Ajilchdiin tsalin bodoh funkts

bool hcounter(float tsag); //Ajillasan tsagiig nemeh

void swap(employee &A, employee &B); //Swap funkts

};

employee::employee() //Baiguulagch funkts

{

cout << "Object amjilttai uuslee\n";

id = 0;

name = new char[4];

strcpy(name, "anh");

pos = new char[7];

strcpy(pos, "worker");

wh = 0;

}

employee::~employee() //Ustgagch funkts

{

delete[] name;

delete[] pos;

cout << "Object amjilttai ustlaa\n";

}

employee::employee(int i, char \*n, char \*p, float w)

{

staffNo = i;

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

strcpy(name, n);

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

strcpy(pos, p);

wh = w;

}

void employee::setId()

{

cout << "id:";

cin >> id;

}

void employee::setName()

{

delete name;

char n[20];

cout << "ner:";

cin >> n;

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

strcpy(name, n);

}

void employee::setName1(char\* str)

{

delete name;

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

strcpy(name, str);

}

void employee::setPos()

{

delete pos;

char p[10];

cout << "alban tushaal:";

cin >> p;

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

strcpy(pos, p);

}

void employee::setPos1(char\* str)

{

delete this->pos;

this->pos = new char[strlen(str) + 1];

strcpy(this->pos, str);

}

void employee::setWh()

{

cout << "tsag:";

cin >> wh;

}

int employee::getId()

{

return id;

}

char \*employee::getName()

{

return name;

}

char \*employee::getPos()

{

return pos;

}

float employee::getWh()

{

return wh;

}

void employee::put()

{

cout << "--------------------------------------------" << endl;

cout << " id || ner || pos || tsag\n";

cout << " " << id << " " << name << " " << pos << " " << wh << endl;

}

float employee::dcal()

{

return 10000 \* wh;

}

float employee::cal()

{

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

return dcal();

else

return 5000 \* wh;

}

bool employee::hcounter(float tsag)

{

if (tsag > 0 && tsag <= 24)

{

wh = wh + tsag;

return true;

}

else

return 0;

}

void swap1(employee &A, employee &B)

{

employee temp;

temp = A;

temp.setName1(A.getName());

temp.setPos1(A.getPos());

A = B;

A.setName1(B.getName());

A.setPos1(B.getPos());

B = temp;

B.setName1(temp.getName());

B.setPos1(temp.getPos());

}

int main()

{

int i,n;

float salary, tsag;

cout << "heden object orulah ve?"<< endl;

cin >> n;

employee A[n];

int l;

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

{

A[i].setId();

A[i].setName();

A[i].setPos();

A[i].setWh();

}

int j;

for(i = 0; i<n-1;i++){

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

if (A[j].cal() < A[j+1].cal()){

swap1(A[j], A[j+1]) ;

}

}

}

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

A[i].put();

}