# Классын хүснэгт болон бусад функцийн талаарх ойлголт

# (Лаборатори №5)

Г. Саруул (20B1NUM2095)

ХШУИС, МКУТ-ийн Компьютерийн ухаан хөтөлбөрийн оюутан

# 1. ОРШИЛ

Ажилтны мэдээллийг авч түүнийг нэрээр ангилах үйлдэл хийсэн. Классын гишүүн өгөгдлийн төрлийг сольсон. Классын array -г эрэмбэлдэг insertionSort нэртэй функц тодорхойлсон.

# 2.ЗОРИЛГО

Олон ажилтны мэдээллийг авч түүнийг хэрэглэгчийн хүссэн байдлаар эрэмбэлэн харуулах

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Байгуулагч функц хэзээ дуудагддаг вэ ?

Байгуулагч функц буюу Байгуулагч нь классаас шинэ объект үүсэх үед хамгийн эхэнд дуудагдана. Классд дамжуулах параметрээс хамаарч ямар төрлийн байгуулагч дуудагдах нь тодорхой болдог. Ямар ч параметргүй классын нэрээр дуудвал анхдагч байгуулагч функцыг дуудна.

3.2 Устгагч функц хэзээ дуудагддаг вэ?

Устгагч функц нь цааш хэрэглэгдэхгүй объект устахад дуудагдаж түүний эзэмшиж байсан динамик ойг чөлөөлж өгдөг. Устгагч функц нь объект устгагдах үед дуудагддаг. Хэрвээ хөгжүүлэгч устгагч функц бичээгүй бол байгуулагч функцийн нэгэн адилаар С++ компайлар түүнийг нэмэж өгдөг ба устгагч функц нь классынхаа гишүүн функц болох бөгөөд функцын нэр нь классын нэртэй ижил ба урдаа ~ тэмдэгт авна. Устгагч функц нь application, программ дуусах үед буюу объектийн үйлчлэх хүрээ дуусах дуудагдана.

3.3Хуулагч байгуулагч гэж юу вэ? Ач холбогдол нь юу вэ?

Хуулагч байгуулагч бол шинэ объект нь түүний өмнөх аль нэгэн объектын сүүлийн утгыг хуулбарлаж авах боломжийг олгох функц юм. Хуулагч байгуулагч нь параметр авна. Тэр параметр нь класс нэгтэй хуулагдах объектын заалт байна. Хуулагч байгуулагч дуудагдах:

a. Объект классын өөр объектийн мэдээлэлтэй ижилхэн үүсгэх үед

b. Объект параметрээр дамжих үед

c. Compiler түр объект үүсгэх үед

3.4 Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн урьдчилан сэргийлэх вэ?

Хэрвээ объект заагч эсвэл санах ой нөөцөлж байгаа бол түүнд зориулж хуулагч функц үүсгэх хэрэгтэй байдаг. Санах ойн цоорхой үүсэх шалтгаан нь хуулагчаар объект а-г объект б-д хуулвал б-д а объектын утга биш хаяг оноогддог. Энгийнээр хэлбэл 2 объект нэг хаягтай болно гэсэн үг. Үүнээс сэргийлэхийн тулд санах ой нөөцөлсөн хуулагч гишүүн функц тодорхойлж өгдөг. Жишээ нь:

Pet::Pet(const Pet& huuchinPet){

ner = new char[strlen(huuchinPet.ner)+1];

strcpy(ner, huuchinPet.ner);

}

3.5Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? New оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

Тухайн объект үүсэж буй классаас хаяган хувьсагч үүсгэе

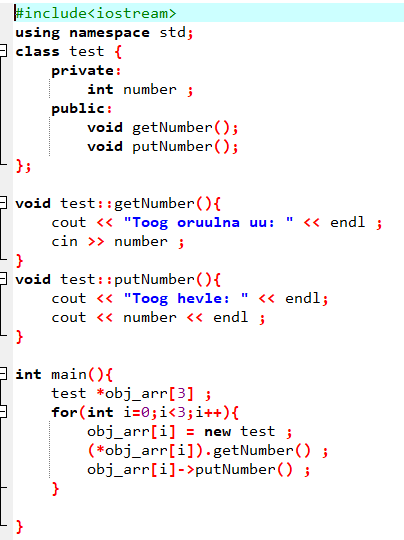
КлассНэр \*ХаяганХувьсагчийнНэр；

Хаяган хувьсагчаа үүсгэсний дараа түүнд харгалзах санах ойг нөөцөлнө. Үүний тулд new түлхүүр үгийн тусламжтайгаар үүсгэж өгнө.

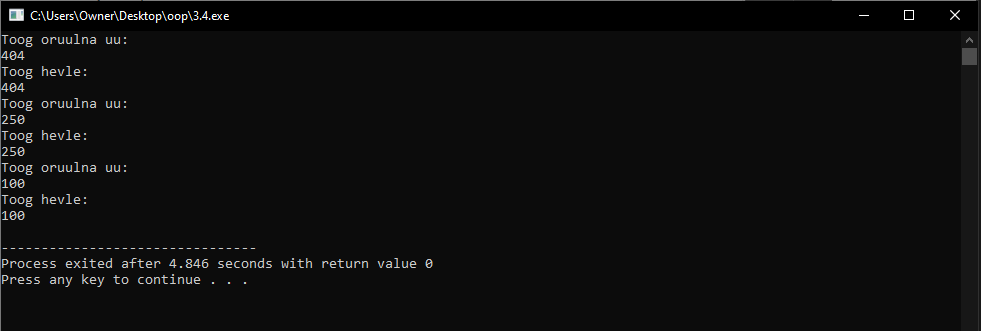
ХаяганХувьсагчийнНэр = new КлассНэр();

Ингэснээр ХаяганХувьсагчийнНэр хаяган хувьсагчид КлассНэр() классын хэмжээтэй динамик санах ойг нөөцөлж өгнө. Түүний гишүүн өгөгдөлд эсвэл функцэд хандахдаа ХаяганХувьсагчийнНэр->гишүүнӨгөгдөл эсвэл ХаяганХувьсагчийнНэр->гишүүнФунц() гэж хандаж болно. Өөрөөр хэлвэл -> операторыг ашиглана. Эсвэл (\*ХаяганХувьсагчийнНэр).гишүүнФункц гэж хандаж болно. Харин санах ой нөөцөлж өгсөн бол түүнийг ашиглаж дуусах үедээ чөлөөлөх үйлдэлийг хийх хэрэгтэй үүнийг delete түлхүүр үгийн тусламжтайгаар хийдэг. delete ХаяганХувьсагчийнНэp;

3.6 Объектын хаяган хувьсагчийн хүснэгт үүсгэх жишээ код бичиж ажиллуул.

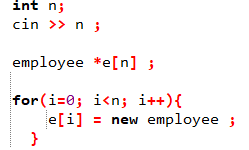


Кодын ажлуулсан нь:

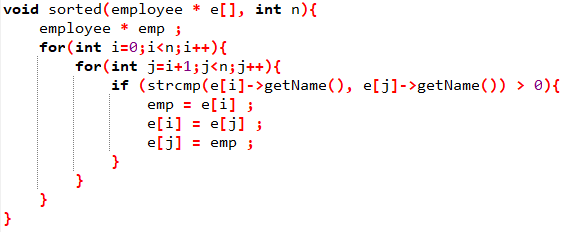


# 4. ХЭРЭГЖҮҮЛЭЛТ

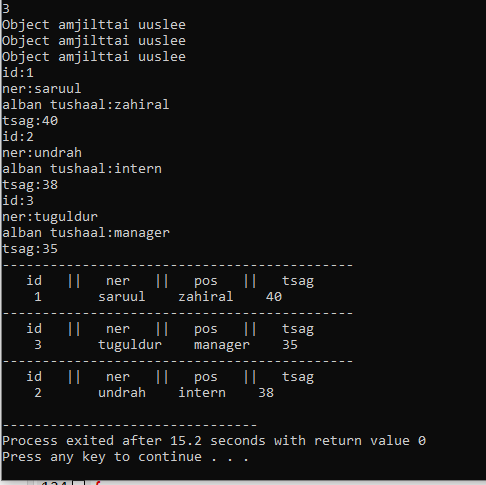
Олон ажилчин бүртгэж ажилчдыг нэрээр нь эрэмбэлэх. Нэрээр эрэмбэлэхдээ объектын хаяган хүснэгт үүсгээд хаягийг нь эрэмбэлбэлнэ.



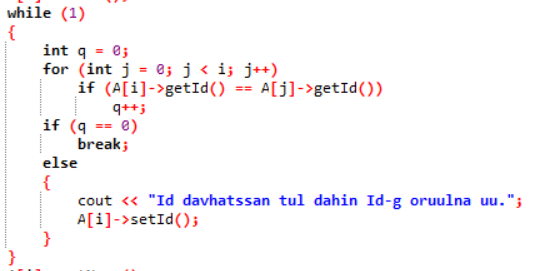
Эрэмбэлэх функц:



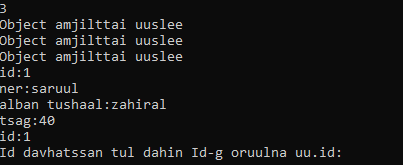
Программыг ажлуулсан үр дүн:



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



Id давхцсан үед ингэж ажиллана:



# 5. ДҮГНЭЛТ

Анхдагч байгуулагч, хуулагч байгуулагч, параметртэй байгуулагч болон бусад дахин тодорхойлсон байгуулагчид нь зөвхөн объект үүсэх үед л дуудагддаг ба үүссэний дараа тэдгээрт хандаж болохгүй. Харин устгагч байгуулагч нь зөвхөн объект устах үед дуудагдах ба түүнийг дахин тодорхойлж болдоггүй. Мөн хуулагч байгуулагч нь санах ойн нөөцөлсөн объектыг хуулахдаа санах ойн цоорхой үүсгэх тул хэрэглэгчийн хуулагч функц бичих замаар шийдвэрлэдэг байна.

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

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::setPos()

{

delete pos;

char p[10];

cout << "alban tushaal:";

cin >> p;

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

strcpy(pos, p);

}

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 employee::swap(employee &A, employee &B)

{

employee temp;

temp = A;

A = B;

B = temp;

}

void sorted(employee \* e[], int n){

employee \* emp ;

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

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

if (strcmp(e[i]->getName(), e[j]->getName()) > 0){

emp = e[i] ;

e[i] = e[j] ;

e[j] = emp ;

}

}

}

}

int check\_id(employee \* e[], employee \* e1, int n){

int i = 0;

bool val = false ;

while(i < n){

if (e[i] != e1){

if (e[i]->getId() == e1->getId()){

val = true ;

}

}

i++ ;

}

return val ;

}

int main()

{

int i;

float salary, tsag;

int n;

cin >> n ;

employee \*e[n] ;

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

e[i] = new employee ;

}

int l;

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

{

e[i]->setId();

while (1)

{

int q = 0;

for (int j = 0; j < i; j++)

if (e[i]->getId() == e[j]->getId())

q++;

if (q == 0)

break;

else

{

cout << "Id davhatssan tul dahin Id-g oruulna uu.";

e[i]->setId();

}

}

e[i]->setName();

e[i]->setPos();

e[i]->setWh();

}

// NEREER EREMBELEH

sorted(e, n) ;

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

e[i]->put();

}