# Байгуулагч, Устгагч Функц (Лаборатори №4)

О. Ихбаяр

ХШУИС, Програм хангамж, 3-р түвшин, 17b1num2575@stud.num.edu.mn

# 1. ОРШИЛ

Байгуулагч, устгагч функц болон динамик санах ой гэж юу болох тэдгээрийг хэрхэн хэрэглэх болон функц дахин тодорхойлолтын талаар авч үзэх болно.

# 2. ЗОРИЛГО

Байгуулагч функц, устгагч функц болон анхдагч байгуулагчийн давуу тал, тэдгээрийн хэрэглээний талаарх ойлголтыг авах.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Байгуулагч функц

Гишүүн өгөгдөлд гарааны утга оноох зорилгоор байгуулагч функцыг хэрэглэдэг. Байгуулагч функц нь классынхаа нэртэй ижил байна.

## 3.2 Устгагч функц

Устгагч функц нь цаашид хэрэглэгдэхгүй болсон объектыг санах ойгоос чөлөөлөх үүрэгтэй функц юм. Класс дотор устгагч функцийг тодорхойлж өгөөгүй тохиолдолд c++ компайлэр автоматаар нэмж өгдөг.

## 3.3 Анхдагч байгуулагч

Анхдагч функц гэдэг нь програм хөгжүүлэгч ямар нэгэн байгуулагч тодорхойлж өгөөгүй байвал С++ компайлер өөрөө ийм функцийг програмд нэмж тодорхойлохыг хэлнэ.

## 3.4 Функц дахин тодорхойлох

Ижил нэртэй бөгөөд параметрых нь төрөл, тоо, дараалал нь өөр функцууд байж болох бөгөөд үүнийг функц дахин тодорхойлох гэнэ.

## 3.5 NEW оператор

Өгөгдсөн төрлөөр санах ойд шинэ ой нөөцлөөд эхлэл хаягийг нь буцаадаг оператор. Нөөцөлсөн санах ойгоо чөлөөлөхгүй бол төхөөрөмж унтартал устахгүй.

## 3.6 Устгагч функц

Нөөцөлсөн санах ойг чөлөөлөх үүрэгтэй, классынхаа нэртэй ижил нэртэй ба урдаа ~ тэмдэгтэй байна.

# 4. ХЭРЭГЖҮҮЛЭЛТ

## 4.1 Устгагч функц

Class employee{

//Устгагч функц тодорхойлох

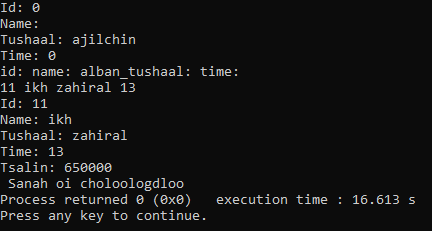
~employee(){

cout<<"\n Sanah oi choloologdloo";

}

}

Автоматаар дуудагдан ажилсан байдал.



## 4.2 Гишүүн функц дотор гишүүн функц дуудах

float employee::salary(){

//Tushaal ni zahiraltai tentsuu esehiig shalgana

int res=strcmp(tushaal,"zahiral");

if(res==0){

// Zahirliin alban tushaaltai uyd zahirliin tsaling bodoh function duudah

return time\*20000+zah\_salary();

}else{

return time\*20000;

}

}

## 4.3 Ажилтан классын хүснэгтийг эрэмбэлэх

## //nootsloh temp object uusgeh

## employee temp;

## //buble sort ermblelt

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

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

## if(emp[j].salary()>emp[j+1].salary()){

## temp=emp[j];

## emp[j]=emp[j+1];

## emp[j+1]=temp;

## }

## }

## }

//erembelsen husnegtee hevleh

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

## cout<<"\t"<<i+1<<"-r ajiltan\n";

## emp[i].print();

}

BUBLE SORT нь хөрш хоёр элементүүдийн байрыг солих замаар бүх элемент нь зөв байраа олох хүртэлх үйлдлийг хийдэг алгоритм юм.

## 4.4 Динамик санах ой болон байгуулагч, устагч функцийг хамтад нь ашиглах

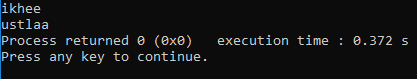
Байгуулагч функц дотроо динамик санах ой үүсгэнэ. Устгагч функц дотроо delete оператороор нөөцөлсөн санах ойгоо чөлөөлж өгнө.

class R{

char \*name;

public:

R(){

 name = new char[8];

strcpy(name,"ikhee");

cout<<name<<"\n";

}

~R(){

delete name;

cout<<"ustlaa";

}

}

# 5. ДҮГНЭЛТ

Гишүүн функц дотор гишүүн функцийг дуудаж болох бөгөөд устгагч функц нь програм ажилаад дуусах үед нөөцөлсөн санах ойг чөлөөлж өгдөг. Объект санах ойгоос чөлөөлөгдөх үед автоматаар дуудагдана.

# 6. АШИГЛАСАН МАТЕРИАЛ

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

# 7. ХАВСРАЛТ

#include<iostream>

#include<string.h>

using namespace std;

//Ajilchin class todorhoiloh

class employee {

//private handaltiin tuvshintei gishuun ogogdol bolon function

private:

int id;

char name[20];

char tushaal[20];

float time;

float zah\_salary();

public:

float salary();

bool increase\_time(float i);

//parametrtei baiguulagch function

Employee(int id, char name[], char tushaal[], int time){

id=id;

int count=0;

//name-d sanah oi nuutsulj baina.

for(int i=0;i<strlen(name);i++){

count++;

}

name=new char(count);

strcpy(name,name);

count=0;

//tushaal-d sanah oi nuutsulj baina.

for(int i=0;i<strlen(tushaal);i++){

count++;

}

tushaal=new char(count);

strcpy(tushaal,tushaal);

time=time;

}

//Baiguulagch fuction todorhoiloh

employee(){

id=1;

time=0;

strcpy(name,"");

strcpy(tushaal,"zahiral");

}

//Ustgagch function todorhoiloh

~employee(){

cout<<"\n Sanah oi choloologdloo";

delete[] name;

delete[] tushaal;

}

//utga oruuulah

void setId(int id2) { id=id2; }

void setName(char name1[]) {

strcpy(name,name1);

}

void settushaal(char tushaal1[]) {

strcpy(tushaal,tushaal1);

}

void setTime(int time1) {

time=time1;

}

//Utga butsaah

int getId() {

return id;

}

char\* getName() {

return name;

}

char\* gettushaal() {

return tushaal;

}

int getTime() {

return time;

}

};

void print(employee \*emp, int i){

cout<<emp[i].getId()<<" "<<emp[i].getName()<<" "<<emp[i].gettushaal()<<" "<<emp[i].getTime()<<" "<<emp[i].salary()<<endl;

}

main(){

int i, n=3,j,bair;

//Array uusgeh

employee emp[n];

//Noots object uusgeh

employee temp;

//Utga oruulah heseg

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

cout<<i+1<<"-r hun:\n"<<"id: name: tushaal: time: \n";

int id, time;

char\* name=new char[20];

char\* tushaal=new char[10];

cin>>id>>name>>tushaal>>time;

emp[i].setId(id);

emp[i].setName(name);

emp[i].settushaal(tushaal);

emp[i].setTime(time);

}

//Bubble sort tsalingaar erembleh

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

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

if(emp[j].salary()>emp[j+1].salary()){

temp=emp[j];

emp[j]=emp[j+1];

emp[j+1]=temp;

}

}

}

//Erembelsnee hevleh

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

cout<<"\t"<<i+1<<"-r ajiltan\n";

print(emp,i);

}

}

float employee::salary(){

int res=strcmp(tushaal,"zahiral");

if(res==0){

return time\*20000+zah\_salary();

}else{

return time\*20000;

}

}

float employee::zah\_salary(){

return 30000\*time;

}

bool employee::increase\_time(float i){

time=time+i;

if(time>=0 && time<=24){

true;

}

false;

}