

(Лаборатори №5)

ХШУИС, МКУТ

Г.Мөнх-Оргил 18b1num2048

1. ОРШИЛ

Объект байгуулахын тулд C++ -н компайлер тодорхой ажил хийдэг, тухайлбал гишүүн өгөгдөлд ой бэлдэж түүнд гарааны утга оноодог бөгөөд үүнтэй адил байгуулагч функцуудыг энэ ажлын хүрээнд авч үзэх болно.

2. ЗОРИЛГО

1. Байгуулагч функц хэзээ дуудагддаг вэ?
2. Устгагч функц хэзээ дуудагдах вэ?
3. Хуулагч байгуулагч гэж юу вэ? Ач холбогдол нь юу вэ?
4. Хуулагч функц гэж юу вэ? Ач холбогдол нь юу вэ? Санах ойн цоорхойгоос хэрхэн сэргийлэх вэ?
5. Объектын хаяган хувьсагчийг хэрхэн зарлах вэ? new оператороор санах ой нөөцлөх, хаяган хувьсагчаар дамжуулж объектын гишүүн өгөгдөл, гишүүн функцэд яаж хандах вэ?

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц хэзээ дуудагддаг вэ?

Объект байгуулагдах үед түүнд зориулан ой бэлдэхээр анхдагч эсвэл хэрэглэгчийн тодорхойлсон байгуулагч функц үргэлж дуудагдана. Тиймээс ямар нэгэн байгуулагч функц дуудахгүйгээр объект байгуулагдахгүй.

3.2 Устгагч функц хэзээ дуудагдах вэ?

Объектэд зориулан байгуулагч функцийг бэлдэх ойг хэрэглэж дууссаны дараагаар чөлөөлөхөд устгагч функцийг хэрэглэнэ. Устгагч функц нь цаашдаа хэрэглэгдэхгүй объект устахад түүний эзэмшиж байсан ойг чөлөөлнө. Хэрэв класс дотор устгагч функцийг илээр тодорхойлж өгөөгүй тохиолдолд объект устахад анхдагч устгагч функцийг C++ компайлер нэмж тодорхойлж хэрэглэнэ.

3.3 Хуулагч байгуулагч гэж юу вэ?

Шинэ объект өмнөх аль нэгэн объектын тухайн үеийн утгыг гарааны утга болгон авахыг хуулагч байгуулагчаар шийдэж болно. Хуулагч байгуулагч нь объект хуулбарлах функц юм.

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

Хуулагч байгуулагч бол шинэ объект нь түүний өмнөх аль нэгэн объектын сүүлийн утгыг хуулбарлаж авах боломжийг олгох функц юм. Ийм маягаар байгуулагдах шинэ объект ба утга нь хуулбарлагдах объект хоорондоо зөвхөн нэрээрээ ялгаатай.

Ийм байгуулагч нэг параметр авах ба тэр нь класс нэртэй хуулагдах объектын заалт байна.

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

объектийн хаяган хувьсагчийг дээрх байдлаар зарлаж болно.

```
employee *pptr;
```

```
employee emp;
```

```
pptr = &emp
```

обектон хаяган хувьсагчаар дамжуулж гишүүн функцэд хандах

- Гишүүнчлэлийн цэг (.) оператор хэрэглэх

```
(*pptr).getdata ();
```

```
(*pptr).showdata ();
```
- Гишүүнчлэлийн сум (→) оператор хэрэглэх

```
pptr→ getdata ();
```

```
pptr→ showdata();
```

```
test* pd = new test(2);
```

New түлхүүр үг шинээр үүсэн объектыг санах ой дээр байрлуулдаг. Энэ нь C хэлний Malloc тай адилхан ажиллаж, дараа нь бидэнд утсгах боломж олгодог.

```
test ob(2);
```

New түлхүүр үг ашиглахгүй шууд зарлах үед объект нь санах ой дээр шууд байрладаггүй ба санах ойтэй холбох ажилгааг компайлер хийдэг. Классын хүрээнээс гарахад устдаг эсвэл мартагддаг.

4. Хэрэгжүүлэлт

1. new оператор ашиглан санах ой нөөцлөх

employee төрлийн класс-д классын нэртэй адил функц үүсгэсэн. Объект үүсэхэд анхны утгууд нь гишүүн өгөгдөл тус бүрийг утгатай байлгахаар зарлаж өгсөнийг доорх жишээнээс харж болно.

```
employee :: employee(){
    id = 0;
    name = new char[2];
    strcpy(name, "");
    position = new char[7];
    strcpy(position, "ajilchin");
    wtime = 0;
    cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<wtime<<endl;
}
```

2. Санах ой чөлөөлөх

Гишүүн өгөгдөлд хоёр char төрлийн хувьсагч байгаа бөгөөд объект үүсэн ашиглагдаад дуусах үед санах ой дээр ойн цоорхой үүсэх учир классын нэртэй адил нэртэй урдаа ~ оператор залган байгуулан, тухайн char төрлийн гишүүн өгөгдөлүүдийг хаягаар нь хандан устгах функц бичсэн. Програм ажиллаж дуусах үед устгагч функц ажиллаж гишүүн өгөгдөлүүдийг устгасанаа хэвлэсэн.

```
employee :: ~employee(){
    delete name, position;
    cout<<"object deleted\n";
}
```

3. Нэрээр эрэмбэлэх

Хүснэгтээр объектуудыг дамжуулж, дамжуулагдсан объектуудын нэрийг эхнээс нь сонгон авж strcmp функц ашиглан хооронд жиших замаар нэрээр эрэмбэлсэн.

```
void sort_name(employee *b[], int h){
    int i, j;
    for(i=0; i<h-1; i++){
        for(j=0; j<h-i-1; j++){
            if(strcmp(b[j]->get_name(), b[j+1]->get_name())==1){
                employee *t;
                t=b[j];
                b[j]=b[j+1];
                b[j+1]=t;
            }
        }
    }
}
```

4. ID шалгах

Хүснэгтээр ажилчдын id-ийг дамжуулж, өмнөх id-тай тэнцүү эсэхийг шалгах замаар

гүйцэтгэсэн.

```
void get_data(employee a[], int h){
    int i, j, id, x;
    float b;
    char name[20], position[15];
    for(i=0; i<h; i++){
        cout<<"Ajilchnii Id: ";
        cin>>id;
        for(j=0; j < i; j++){
            if(id==a[j].get_id()){
                cout<<"ID давхардсан";
                exit(1);
            }
        }
    }
}
```

5. ДҮГНЭЛТ

Энэхүү лабораторийн ажлаар байгуулагч функц, устгагч функц гэж юу болох, эдгээрийг хэрхэн хэрэглэх талаар судалж мэдлээ. Байгуулагч функц тодорхойлсноор классын объектод санаах ой нөөцлөх, гарааны утга оноох ажлуудыг хийж болох ба ямар нэгэн байгуулагч функц дуудагдахгүйгээр объект байгуулагдахгүй, үүнтэй адилаар объект нь устгагч функцгүйгээр устаж чадахгүй.

6. Ашигласан материал

1. <https://www.geeksforgeeks.org/constructors-c/>
2. https://www.tutorialspoint.com/cplusplus/cpp_constructor_destructor.html
3. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар

7. Хавсралт

```
#include<iostream>

#include<string.h>

#include<conio.h>

#include<windows.h>

#include<bits/stdc++.h>

using namespace std;

class employee{
```

private:

```
    int id;  
    char *name;  
    char *position;  
    float worked_time;  
    int zahirliin_tsalin();
```

public:

```
    employee();  
    employee(int a, char *b, char *c, float d);  
    void set_id(int a);  
    int get_id();  
    void set_name(char *a);  
    char* get_name();  
    void set_position(char *a);  
    char* get_position();  
    void set_worked_time(float a);  
    float get_worked_time();  
    int tsalin_bodoh();  
    ~employee();
```

};

employee :: employee(){

```
    id = 0;  
    name = new char[2];  
    strcpy(name, "");  
    position = new char[7];  
    strcpy(position, "ajilchin");
```

```

        worked_time = 0;

        cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<worked_time<<endl;
    }

    employee :: employee(int a, char *b, char *c, float d){
        id=0;

        name = new char (strlen(name));

        strcpy(name, b);

        position = new char(strlen(position));

        strcpy(position, c);

        worked_time = 0;

        cout<<id<<"\t"<<name<<"\t"<<position<<"\t"<<worked_time<<endl;
    }

    employee :: ~employee(){
        delete name, position;

        cout<<"object deleted\n";
    }

    void employee :: set_id(int a){
        id=a;
    }

    int employee :: get_id(){
        return id;
    }

    void employee :: set_name(char *a){
        name = new char(strlen(a)+1);

        strcpy(name, a);
    }

```

```
}
```

```
char* employee :: get_name(){
```

```
    return name;
```

```
}
```

```
void employee :: set_position(char *a){
```

```
    position = new char(strlen(a)+1);
```

```
    strcpy(position, a);
```

```
}
```

```
char* employee :: get_position(){
```

```
    return position;
```

```
}
```

```
void employee :: set_worked_time(float a){
```

```
    worked_time=a;
```

```
}
```

```
float employee :: get_worked_time(){
```

```
    return worked_time;
```

```
}
```

```
int employee :: zahirliin_tsalin(){
```

```
    return worked_time*22*5000;
```

```
}
```

```

int employee :: tsalin_bodoh(){
    int a, b;
    if(strcmp(position, "zahiral")!=0){
        a=worked_time*22;
        b=a*3750;
        return b;
    }
    else
        return zahirliin_tsalin();
}

```

```

void get_data(employee a[], int h){
    int i, j, id, x;
    float b;
    char name[20], position[15];
    for(i=0; i<h; i++){
        cout<<"Ajilchnii Id: ";
        cin>>id;
        for(j=0; j < i; j++){
            if(id==a[j].get_id()){
                cout<<"ID davhardsan";
                exit(1);
            }
        }
        a[i].set_id(id);
        cout<<"Ajilchnii ner: ";
        cin>>name;
    }
}

```



```

        a[i].set_name(name);
        cout<<"Ajilchnii alban tushaal: ";
        cin>>position;
        a[i].set_position(position);
        cout<<"Ajilchnii ajillasan tsag: ";
        cin>>b;
        a[i].set_worked_time(b);
    }
}

```

```

void print(employee a[], int h){
    int i;
    for(i=0; i<h; i++){
        cout<<a[i].get_id()<<"\t";
        cout<<a[i].get_name()<<"\t";
        cout<<a[i].get_position()<<"\t\t";
        cout<<a[i].get_worked_time()<<"\t";
        cout<<a[i].tsalin_bodoh()<<"\n";
    }
}

```

```

void sort_name(employee *b[], int h){
    int i, j;
    for(i=0; i<h-1; i++){
        for(j=0; j<h-i-1; j++){
            if(strcmp(b[j]->get_name(), b[j+1]->get_name())==1){
                employee *t;

```

```

        t=b[j];
        b[j]=b[j+1];
        b[j+1]=t;
    }
}

cout<<"sorted name: "<<endl;
for(i=0; i<h; i++){
    cout<<b[i]->get_id()<<"\t";
    cout<<b[i]->get_name()<<"\t";
    cout<<b[i]->get_position()<<"\t\t";
    cout<<b[i]->get_worked_time()<<"\t";
    cout<<b[i]->tsalin_bodoh()<<"\n";
}
}

```

```

int main(){
    int h;
    cout<<"Heden ajilchin burtgeh ve: "<<endl;
    cin>>h;
    employee a[h];
    get_data(a, h);
    employee *b[h];
    int i;
    for(i=0; i<h; i++)

```

```
        b[i]=&a[i];  
    sort_name(b, h);  
}
```