



Цэрэнбат Номин-Эрдэнэ
(20B1NUM0426)

ICSI201 - Объект хандлагат програмчлал

МУИС-ХШУИС
Мэдээллийн технологи

Лаборатори №4

Цэрэнбат Номин-Эрдэнэ

1. ОРШИЛ

Байгуулагч функц, устгагч функц, дахин тодорхойлогдсон функц болон динамик санах ойн new, delete операторуудын талаарх ойлголтоо бататган классын хүснэгт ашиглан бодлого бодно. Мөн лаборатори 3д тодорхойлсон классыг ашиглан өгөгдсөн даалгаврыг хийж гүйцэтгэсэн.

2. ЗОРИЛГО

- Лабораторын ажлыг алдаагүй зөв хийж гүйцэтгэх.
- Байгуулагч функц, Устгагч функц, Функц дахин тодорхойлох тухай уншиж судлах
- Динамик санах ой (new, delete оператор ашигласан) болон байгуулагч, устгагч функцийг хэрхэн хамтад нь ашиглаж сурах
- Lab03-д тодорхойлсон класст анхдагч болон параметртэй байгуулагч нэмж тодорхойлох
- Lab03-д тодорхойлсон класст устгагч функц тодорхойлж устгагч функц хэзээ дуудагдаж байгааг туршилтад үндэслэн хариулах
- Lab03-д тодорхойлсон класст гараас өгөгдсөн түлхүүр үг орсон бүх судлаачдын өгүүллийг хэвлэх статик public функц хэрэгжүүлэх.

3. ОНОЛЫН СУДАЛГАА

3.1 Байгуулагч функц гэж юу вэ?

Байгуулагч гэдэг нь тухайн классын объектуудыг эхлүүлэх классын гишүүн функц юм. С++ дээр объект (класс жишээ) үүсгэх үед байгуулагч функц автоматаар дуудагддаг. Энэ бол классын тусгай гишүүн функц юм. Байгуулагч нь ердийн функцээс дараах байдлаараа ялгаатай байдаг. Байгуулагч функц нь класстай ижил нэртэй байдаг. Байгуулагчид буцах төрөл байдаггүй. Объект үүсгэх үед байгуулагч нь автоматаар дуудагддаг. Хэрэв байгуулагчийг заагаагүй бол С++ хөрвүүлэгч нь бидний хувьд анхдагч утгыг үүсгэдэг.(ямар ч параметр хүлээхгүй, хоосон биетэй байдаг).

Анхдагч байгуулагч нь аргументгүйгээр дуудаж болох байгуулагч юм (хоосон параметрийн жагсаалтаар тодорхойлогдсон эсвэл параметр бүрт өгөгдсөн анхдагч

аргумент бүхий). Тогтмол эсвэл лавлагаа хэлбэрийн гишүүдтэй классын хувьд анхдагч байгуулагч үүсгэгддэггүй.

3.2 Устгагч функц гэж юу вэ?

Устгагч функц гэдэг нь объектын динамик гишүүн өгөгдлийн санах ойгоос бүрэн чөлөөлөгдөхөд хэрэглэдэг. Устгагч функцэд ямар ч argument үүсгэдэггүй бөгөөд юу ч буцаадаггүй. Хэрэв класс устгагч функцээр хангагдаагүй бол объектийн мужаас гаргахад C++ хэл нь өөрийн устгагч функцийг автоматаар дуудаж объектийн гишүүдийн ашигласан санах ойг чөлөөлдөг.

Устгагч функцийн шинж чанар:

- Объектууд устгагдсан тохиолдолд устгагчийн функц автоматаар дуудагддаг.
- Үүнийг статик эсвэл `const` гэж зарлах боломжгүй.
- Энэ нь хүчингүй ч гэсэн буцах төрөлгүй.
- Устгагчтай классын объект нь гишүүн функц болж чадахгүй.
- Классын нийтийн хэсэгт устгагчийг зарлах хэрэгтэй.
- Программист нь устгагч хаяг руу нэвтрэх боломжгүй. Устгагчийг хэзээ дууддаг вэ? Объект хамрах хүрээнээс гарах үед устгах функцийг автоматаар дууддаг.

(1) функц дуусах үед

(2) програм дуусах үед

(3) локал хувьсагч агуулсан блок дуусах үед

(4) устгах операторыг дууддаг

3.3 Функц дахин тодорхойлох гэж юу вэ? Жишээгээр тайлбарал.

C/C++ хэлэнд ижил нэртэй боловч авах аргументийн төрөл, дараалал, тоо нь ялгаатай байх олон янзын функцыг тодорхойлох боломжтой. Ингэж нэр, төрөл нь ижил олон функц тодорхойлох үйлдлийг функц дахин тодорхойлох гэнэ. Жишээлбэл: `int a = 5; int a = 6; Мөн. int foo ();` тодорхойлолт биш юм.

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

Мэдээллийн технологид гурван төрлийн санах ой байдаг ба тэдгээр нь:

1. Статик – Програм ажиллаж эхлэмэгц санах ойг нөөцөлдөг
2. Локал – Функц дуудагдахад стекд санах ой нөөцөлдөг ба функц дуусмагц санах ойг нь чөлөөлдөг
3. Динамик – Шаардлагатай үед нь санах ой нөөцөлж өгдөг ба програмын хэлнээс нь хамааран устгагч функц дуудагдсан эсвэл програм зогссон үед тухайн санах ойг чөлөөлнө.

Байгуулагч функц дотроо new операторыг ашиглан санах ойгоо нөөцөлж тухайн объект хэрэгцээгүй болмогц 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.

Байгуулагч функц:

```
1  #include<iostream>
2  #include<stdio.h>
3  #include<conio.h>
4  #include <cstring>
5
6  using namespace std;
7
8  class sudlaach {
9      private:
10         int id;
11         char name[20];
12         int age;
13         char degree[20];
14         char * oguulel[10];
15         int oguulel_too;
16
17     public:
18         sudlaach(void);
19         void garaanii_utga();
20         void read();
21         void print();
22         void add_oguulel(char n[]);
23         void search_oguulel(char n[]);
24
25     };
26
```

Параметртэй байгуулагч:

```
1  #include<iostream>
2  #include<stdio.h>
3  #include<conio.h>
4  #include <cstring>
5
6  using namespace std;
7
8  class sudlaach {
9      private:
10         int id;
11         char name[20];
12         int age;
13         char degree[20];
14         char * oguulel[10];
15         int oguulel_too;
16
17     public:
18         sudlaach(int a, char b, int c, char d, char e);
19         sudlaach();
20         void garaanii_utga();
21         void read();
22         void print();
23         void add_oguulel(char n[]);
24         void search_oguulel(char n[]);
25
26 };
```

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

```
92
93  sudlaach::~~sudlaach(){
94      delete []name;
95      cout<<"obyekt ustlaa\n";
96
97  }
```

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

```

public:
    sudlaach(int idnum, char n, int a, char d){
        id = idnum;
        name[20] = n;
        age = a;
        degree[20] = d;
    }
    sudlaach(){
        id = 0;
        strcpy(name, " ");
        age = 28;
        strcpy(degree, "Doctor");
        oguulel_too = 0;
    }

    void setIdnumber(int idnum){
        id = idnum;
    }
    void setName(char n){
        name[20] = n;
    }
    void setAge(int a){
        age = a;
    }
    void setDegree(char d){
        degree[20] = d;
    }
    int const getIdnumber(){
        return id;
    }
    char const getName(){
        return name[20] ;
    }
    int const getAge(){
        return age;
    }
    char const getDegree(){
        return degree[20];
    }
}

1  #include<iostream>
2  #include<stdio.h>
3  #include<conio.h>
4  #include <cstring>
5
6  using namespace std;
7
8  class sudlaach {
9      private:
10         int id;
11         char name[20];
12         int age;
13         char degree[20];
14         char * oguulel[10];
15         int oguulel_too;
16

```

4.5 Судлаач классын хүснэгт үүсгээд гараас хэд хэдэн судлаачдын утга онооно.

```

1  int main(){
2      char oguulel1[10] = "oguulel1";
3      char oguulel2[10] = "oguulel2";
4      char oguulel3[10];
5
6      int i, n;
7      sudlaach s;
8      cout<<"\nHeden sudlaach oruulhaa bichne uu: ";
9      cin>>n;
10     sudlaach obj[n];
11     for(i=0; i<n; i++){
12         obj[i].garaanii_utga();
13         obj[i].read();
14     }
15 }

C:\Users\nomin\OneDrive\Documents\oop\test.exe
Heden sudlaach oruulhaa bichne uu: 2
ID: 123
Name: asd
Age: 18
Degree: doctor
ID: 124
Name: aaa
Age: 19
Degree: doctor

```

4.6 Утга оноосон хүснэгтийг судлаачдын өгүүллийн тоогоор нь эрэмбэлнэ.

7. ХАВСРАЛТ:

```
#include<iostream>
```

```
#include<stdio.h>
```

```
#include<conio.h>
```

```
#include <cstring>
```

```
using namespace std;
```

```
class sudlaach {
```

```
    private:
```

```
        int id;
```

```
        char name[20];
```

```
        int age;
```

```
        char degree[20];
```

```
        char * oguulel[10];
```

```
        int oguulel_too;
```

public:

```
sudlaach(int idnum, char n, int a, char d){
```

```
    id = idnum;
```

```
    name[20] = n;
```

```
    age = a;
```

```
    degree[20] = d;
```

```
}
```

```
sudlaach(){
```

```
    id = 0;
```

```
    strcpy(name, " ");
```

```
    age = 28;
```

```
    strcpy(degree, "Doctor");
```

```
    oguulel_too = 0;
```

```
}
```

```
void setIdnumber(int idnum){
```

```
    id = idnum;
```

```
}
```

```
void setName(char n){
```

```
    name[20] = n;
```

```
}
```

```
void setAge(int a){
```

```
    age = a;
```



```
}  
  
void setDegree(char d){  
  
degree[20] = d;  
  
}  
  
int const getIdnumber(){  
  
return id;  
  
}  
  
char const getName(){  
  
return name[20] ;  
  
}  
  
int const getAge(){  
  
return age;  
  
}  
  
char const getDegree(){  
  
return degree[20];  
  
}  
  
  
  
void add_oguulel(char n[]);  
  
void search_oguulel(char n[]);  
  
~sudlaach();  
  
void print();
```

```
};
```

```
void sudlaach::add_oguulel(char n[]) {  
    if (oguulel_too < 10) {  
        oguulel[oguulel_too] = new char[strlen(n) + 1];  
        oguulel[oguulel_too] = n;  
        oguulel_too++;  
    } else {  
        cout << "Oguulel duursen." << endl;  
    }  
}
```

```
sudlaach::~~sudlaach(){  
    delete []name;  
    cout<<"obyekt ustlaa\n";  
  
}
```

```
void print(){  
    sudlaach s1,s2;  
    cout <<"\nsudlaach1-iin id: "<<s1.getIdnumber();
```

```

        cout <<"\nsudlaach1-iin ner: "<<s1.getName();

        cout <<"\nsudlaach1-iin nas: "<<s1.getAge();

        cout <<"\nsudlaach1-iin degree: "<<s1.getDegree();


        cout <<"\nsudlaach2-iin id: "<<s2.getIdnumber();

        cout <<"\nsudlaach2-iin ner: "<<s2.getName();

        cout <<"\nsudlaach2-iin nas: "<<s2.getAge();

        cout <<"\nsudlaach2-iin degree: "<<s2.getDegree();

    }

```

```

int main(){

    char uguulel1[10] = "uguulel1";

    char uguulel2[10] = "uguulel2";

    char uguulel3[10] = "aaa";


    int i, n, a, idnum;

    char n[20], d[20];


    sudlaach s1, s2;

    /// sudlaach 1

    cout <<"\n sudlaach1-iin id oruulna uu ";

    s1.setIdnumber(id);

```

```
cout <<"\n sudlaach1-iin neriig oruulna uu ";  
  
s1.setName(n);  
  
cout <<"\n sudlaach1-iin nasiig oruulna uu ";  
  
s1.setAge(a);  
  
cout <<"\n sudlaach1-iin degree oruulna uu ";  
  
s1.setDegree(d);  
  
s1.add_oguulel(oguulel1);  
  
s1.add_oguulel(oguulel2);
```

```
/// sudlaach 2
```

```
cout <<"\n sudlaach2-iin id oruulna uu ";  
  
s2.setIdnumber(idnum);  
  
cout <<"\n sudlaach2-iin neriig oruulna uu ";  
  
s2.setName(n);  
  
cout <<"\n sudlaach2-iin nasiig oruulna uu ";  
  
s2.setAge(a);  
  
cout <<"\n sudlaach2-iin degree oruulna uu ";  
  
  
s2.add_oguulel(oguulel3);
```

```
s1.print();
```

