



Цэрэнбат Номин-Эрдэнэ
(20B1NUM0426)

ICSI201 - Объект хандлагат програмчлал

МУИС-ХШУИС
Мэдээллийн технологи

Лаборатори №3

Цэрэнбат Номин-Эрдэнэ

1. ОРШИЛ

Энэ лабораторийн ажил нь класс гэж юу болох түүний онцлог, үүргийн талаар болон C++ объектын ялгаа, гишүүн функц, гишүүн өгөгдлийн хамаарал, өгөгдлийн битүүмжлэл гэж юу болох талаар судлаж ойлгосны үндсэн дээр онолын мэдлэг дээр тулгуулан даалгаврыг хийж гүйцэтгэсэн болно.

2. ЗОРИЛГО

- Лабораторийн ажлыг алдаагүй зөв хийж гүйцэтгэх.
- Класс гэж юу болох онцлог үүргийн талаар судлаж классын талаар мэдлэгтэй болох
- Класс болон C++ объектийн ялгааг олох
- Гишүүн функц, гишүүн өгөгдөл хоёр ямар хамааралтай байдаг, өгөгдлийн битүүмжлэл гэж юу болох
- Классын гишүүн өгөгдөл болон гишүүн функцэд хэрхэн хандах талаар судлаж ойлгосны үндсэн дээр энэхүү лабораторын ажилыг хийж гүйцэтгэсэн болно.

3. ОНОЛЫН СУДАЛГАА

3.1 Класс гэж юу болох, онцлог, үүргийн талаар бич.

Класс гэдэг нь хоорондоо логикийн хувьд уялдаа холбоотой хувьсагчид болон тэдгээр дээр үйлдэл хийх функцуудыг агуулсан хэрэглэгчийн тодорхойлсон төрөл юм. Класс төрөлтэйгээр зарлагдсан хувьсагчыг тухайн классын объект гэж нэрлэдэг. Классын объект бүр бусад объектуудаас ялгагдах дахин давтагдашгүй шинж чанартай байх бөгөөд өөр өөрийн гэсэн санах ойн мужтай байдаг. Классын гишүүн өгөгдөл ба гишүүн функцэд хандахдаа тухайн классын объектын нэрийг цэгтэй хамт бичдэг. Тухайлбал: Энд `b1.setName("English grammar");` мөр нь `b1` объектод зориулан түүний `setName()` гишүүн функцыг дуудаж байна. Нэг ёсондоо `b1` бол бэхлэгдсэн объект юм. Классын гишүүдийг мөн статикаар тодорхойлж болно. Статик гишүүдийн хэрэглээг дараа тодорхой жишээн дээр авч үзэх болно.

Объект хандалтат програмчлалд анги нь объект үүсгэх, төлөв (гишүүн хувьсагчууд) болон зан үйлийн хэрэгжилт (гишүүдийн функцууд эсвэл аргууд) - ын анхны утгыг өгөх өргөтгөх програм-код загвар юм. Олон хэл дээр ангийн нэрийг тухайн ангийн нэр (загвар өөрөө), тухайн ангийн анхдагч байгуулагчийн нэр (объект үүсгэдэг дэд програм), мөн ангийг өдөөж бий болгосон объектын төрөл болгон ашигладаг. эдгээр ялгаатай ойлголтуудыг

хооронд нь амархан уялдуулж болно. Хэдийгээр полиморфик шинж чанартай тул эдгээр хэлнүүд нь яагаад ийм хүчирхэг, динамик, хэрэглэхэд дасан зохицох чадвартай байдаг вэ? Тиймээс тэд динамик системийг (жишээ нь бодит ертөнц, машин сурах, оюун ухаан) илүү хялбар загварчилж чаддаг.

Тухайн объектыг тухайн ангийн байгуулагчаар бүтээх үед үүссэн объектыг тухайн ангийн жишээ гэж нэрлэдэг бөгөөд тухайн объектод хамааралтай гишүүн хувьсагчуудыг анги даяар хуваалцсан ангийн хувьсагчтай харьцуулж, жишээ хувьсагч гэж нэрлэдэг.

Зарим хэл дээр ангиуд нь зөвхөн хөрвүүлэх үеийн шинж чанартай байдаг (шинэ ангиуд ажиллаж байх үед зарлах боломжгүй), харин бусад хэл дээр ангиуд нь нэгдүгээр зэрэглэлийн иргэд байдаг бөгөөд ерөнхийдөө өөрсдөө объектууд байдаг (ихэвчлэн Class төрлийн эсвэл үүнтэй төстэй). Эдгээр хэл дээр анги үүсгэдэг ангиудыг метакласс гэж нэрлэдэг.

- Классыг тодорхойлохдоо class түлхүүр үгийг ашиглана.
- Классын бие нь { } ; хаалтад байрлана.

Классын бие дотор

{

private:

public: гэсэн түлхүүр үгийг ашиглан гишүүний хандах эрхийг тогтооно.

Хандах эрх бичээгүй бол private гэж үзнэ

Ерөнхийдөө гишүүн өгөгдлийг- **private** ,гишүүн функцийг –**public** хандах эрхтэй хэсэгт бичдэг.

};

Гишүүдийн хандах эрх

:public

- Классын гаднаас шууд хандаж болно. Нийтийн хандалт гэх ба хаалттай гишүүдийг гадаад хэсэгтэй холбох үүргийг гүйцэтгэнэ.

:private

- Зөвхөн классын гишүүн функцууд хандана ! Зөвхөн классын дотоод хэсэгт хэрэглэгдэнэ.

3.2 Класс болон C++ объектын ялгаа?

Класс нь өгөгдөл, түүнийг боловсруулах арга бүхий бүтэц. Класс бол тухайн жишээг, өөрөөр хэлбэл объект үүсгэх боломжтой зураг төсөл юм. Объект гэдэг нь програмистуудад анги дотроос хувьсах хэмжигдэхүүн, аргуудыг ашиглахад тусалдаг ангийн жишээ юм. Класс нь нь өгөгдөл болон аргуудыг нэг нэгж болгон холбоход хэрэглэгддэг. Объект нь ангийн хувьсагч шиг ажилладаг.

Класс нь бүлэг хувьсагч ба аргуудыг багтаасан байгууламж гэж үзэж болох объект хандалтат програмчлалын чухал хэсгээс анги ба объектууд; харин объект нь тухайн ангийн гишүүн эсвэл жишээний үүрэг гүйцэтгэдэг.

Класс нь өгөгдөл ба түүнтэй холбоотой функцуудыг хооронд нь холбоход хэрэглэгддэг арга юм. Объект ба анги нь гол төлөв объект хандалтат програмчлалд ашиглагддаг нэр томъёо бөгөөд үүгээр өгөгдлийг код хэлбэрээр, програмын төвөгтэй байдлыг багасгах зорилгоор объект хэлбэрээр зохион байгуулдаг. Объект хандалтат програмчлалд class нь объект үүсгэх, төлөв (гишүүн хувьсагчууд) болон зан үйлийн хэрэгжилт (гишүүдийн функцууд эсвэл аргууд) -ын анхны утгыг өгөх өргөтгөх програм-код загвар юм.

Объектуудын хоорондын харилцааг ашиглан программ бичих программчлалын хандлага (парадигм)-ыг ОХП гэдэг. Өгөгдөл, арга (methods буюу функц)-аас бүрдэх хэрэглэгчийн тодорхойлсон төрлийг класс гэнэ.

3.3 Гишүүн функц, гишүүн өгөгдөл хоёр ямар хамааралтай байдаг вэ? Өгөгдлийн битүүмжлэл гэж юу вэ?

Гишүүн функц нь классынхаа гишүүн өгөгдөл дээр боловсруулалт хийж програмын бусад функцтэй мэдээлэл солилцоно. Гишүүний функцийг нэрлэх конвенц: Функцийн нэр нь үйл үг, эсвэл хэд хэдэн үгнээс бүтсэн үйл үгийн хэллэг байна. Эхний үг нь жижиг үсгээр, үлдсэн үгс нь том үсгээр бичигдсэн (тэмээний хэрэг). Жишээлбэл, `getRadius ()`, `getParameterValues ()`.

Гишүүн өгөгдөл бүтэц гишүүн өгөгдөлтэй байдаг шиг класс бас гишүүн өгөгдөлтэй байна. Өгөгдлийн гишүүний нэр нь нэр үг (статик шинж чанарыг илэрхийлдэг), харин функцын нэр бол үйл үг (үйлдлийг илэрхийлдэг). Тэд ижил нэршилтэй байдаг. Функцууд нь аргументийг хаалтанд авдаг (хоосон хаалтанд тэг аргумент байж болно), харин хувьсагчид үүнийг хийдэггүй. Энэ бичвэрт функцуудыг хос хаалтаар тэмдэглэсэн, жишээлбэл `println ()`, `getArea ()` -ыг тодорхой болгох үүднээс тэмдэглэсэн болно

Өгөгдлийн битүүмжлэл- Класс нь объектийнхоо загвар болдог. Объектийн өгөгдөлд зөвхөн гишүүн функцээр дамжуулж ханддаг. Объектийн өгөгдөлд тухайн классын биш функцээр хандаж болохгүй. Өгөгдөл түүн дээр хийх үйлдэл хийх функцийг хамтад нь холбоотойгоор

авч үздэг нь объект хандлагат технологийн нэгэн чухал шинж юм. Ерөнхийдөө өгөдөл ба функц нь хамтдаа класс гэдэг нэгэн цогц ойлголтонд нэгддэг. Иймд битүүмжлэл бол функц, өгөгдлийн тодорхой түвшинд нээлттэй биш объектийн хүрээнд хамтад нь холбоотойгоор авч үздэг ойлголт юм.

3.4 Классын гишүүн өгөгдөл болон гишүүн функцэд хэрхэн хандах вэ?

Классын гишүүн өгөгдөлд гишүүн функцаар дамжуулан хандана. Мөн классын бүх өгөгдлийг public түвшинд тодорхойлж өгсөн бол объектоор нь дамжуулан хандаж болно. Хандалтын 3 түвшин байдаг:

- Private
- Public
- Protected

Private, public тусгай дохио үгс нь классын гишүүдын үйлчлэх хүрээг заана. Private гишүүд рүү зөвхөн тухайн класс дотроос буюу зөвхөн гишүүн функцаас нь хандаж болно. Харин public гишүүд рүү классын гаднаас, тухайлбар өөр классаас эсвэл өөр функцаас хандаж болно. Protected түвшин нь тухайн классынхаа түвшинд үйлчлэх бөгөөд private түвшинтэй ижил юм.

4. ХЭРЭГЖҮҮЛЭЛТ

4.1 Судлаач гэсэн класс тодорхойлно. Өгөгдсөн түлхүүр үг орсон өгүүллүүд хэвлэнэ үү.

+Дугаар : int

+Нэр : char[20]

+Нас : int

+Эрдмийн зэрэг : char[10]

+Өгүүллүүдийн жагсаалт: char* [10] // Дээд тал нь 10 өгүүллийн жагсаалт агуулах боломжийг олгож байна.

+Гарааны утга оноох() : void // дугаар=0, нэр="", Эрдмийн зэрэг="Доктор", нас = 28

+Гараас утга авах() : void

+Мэдээлэл дэлгэцлэх() : void

+Өгүүлэл нэмэх(char*) : void

+Өгүүлэл хайх(char*) : void // Оролтоор орж ирсэн тэмдэгт мөрийг өгүүллийн нэрсээс хайгаад олдсон өгүүллүүдийг дэлгэцэнд хэвлэх функц

```
3.1.cpp test.cpp
1  #include<iostream>
2  #include<stdio.h>
3  #include<conio.h>
4  #include <cstring>
5
6  using namespace std;
7
8  class sudlaach {
9      public:
10         int id;
11         char name[20];
12         int age;
13         char degree[20];
14         char * oguulel[10];
15         int oguulel_too;
16
17         sudlaach(void); //garaani utga onoono.
18         void read();
19         void print();
20         void add_oguulel(char n[]);
21         void search_oguulel(char n[]);
22
23     };
24
25     sudlaach::sudlaach() {
26         id = 0;
27         strcpy(name, " ");
28         age = 28;
29         strcpy(degree, "Doctor");
30         oguulel_too = 0;
31
32     }
```

```
34 void sudlaach::read() {
35     cout << "\n ID: ";
36     cin >> id;
37     cout << "\n Name: ";
38     cin >> name;
39     cout << "\n Age: ";
40     cin >> age;
41     cout << "\n Degree: ";
42     cin >> degree;
43
44 }
45
46 void sudlaach::print() {
47     cout << "\n ID: " << id;
48     cout << "\n Name: " << name;
49     cout << "\n Age: " << age;
50     cout << "\n Degree: " << degree;
51     cout << "\n Oguulel: \n";
52
53     int i;
54     for (i = 0; i < oguulel_too; i++) {
55         cout << "\t" << oguulel[i] << endl;
56     }
57
58 }
59
60 void sudlaach::add_oguulel(char n[]) {
61     if (oguulel_too < 10) {
62         oguulel[oguulel_too] = new char[strlen(n) + 1];
63         oguulel[oguulel_too] = n;
64         oguulel_too++;
65     } else {
66         cout << "Oguulel duursen." << endl;
67     }
68 }
69
```

3.1.cpp

test.cpp

```
67     }
68 }
69
70 void sudlaach::search_oguulel(char n[]) {
71     int i;
72
73     for (i = 0; i < oguulel_too; i++) {
74         if (strcmp(n, oguulel[i]) == 0) {
75             cout << "\t" << oguulel[i] << endl;
76         }
77     }
78 }
79
80 int main(){
81     char oguulel1[10] = "oguulel1";
82     char oguulel2[10] = "oguulel2";
83     char oguulel3[10];
84
85
86     sudlaach s;
87     s.read();
88     cout << "\n oguulel oruulna uu: ";
89     cin >> oguulel3;
90     s.add_oguulel(oguulel1);
91     s.add_oguulel(oguulel2);
92     s.add_oguulel(oguulel3);
93
94     s.print();
95
96     cout << "\n Haigaad oldson oguulel: \n";
97     s.search_oguulel(oguulel1);
98     s.search_oguulel(oguulel2);
99     s.search_oguulel(oguulel3);
100
101 }
```


C:\Users\nomin\OneDrive\Documents\oop\test.exe

ID: 2021

Name: nomin

Age: 18

Degree: bachelor

oguulel oruulna uu: test123

ID: 2021

Name: nomin

Age: 18

Degree: bachelor

Oguulel:

oguulel1

oguulel2

test123

Haigaad oldson oguulel:

oguulel1

oguulel2

test123

Process exited after 12.1 seconds with return value 0
Press any key to continue . . .

5. XABCPAJIT

```
#include<iostream>
```

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

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

```
#include <cstring>
```

```
using namespace std;
```

```
class sudlaach {
```

```
    public:
```

```
        int id;
```

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

```
        int age;
```

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

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

```
        int oguulel_too;
```

```
        sudlaach(void); //garaani utga onono.
```

```
        void read();
```

```
        void print();
```

```
        void add_oguulel(char n[]);
```

```
        void search_oguulel(char n[]);
```

```
};
```

```
sudlaach::sudlaach() {
```

```
    id = 0;
```

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

```
    age = 28;
```

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

```
    oguulel_too = 0;
```

```
}
```

```
void sudlaach::read() {
```

```
    cout << "\n ID: ";
```

```
    cin >> id;
```

```

    cout << "\n Name: ";
    cin >> name;
    cout << "\n Age: ";
    cin >> age;
    cout << "\n Degree: ";
    cin >> degree;

}

void sudlaach::print() {
    cout << "\n ID: " << id;
    cout << "\n Name: " << name;
    cout << "\n Age: " << age;
    cout << "\n Degree: " << degree;
    cout << "\n Oguulel: \n";

    int i;
    for (i = 0; i < uguulel_too; i++) {
        cout << "\t" << uguulel[i] << endl;
    }

}

void sudlaach::add_uguulel(char n[]) {
    if (uguulel_too < 10) {
        uguulel[uguulel_too] = new char[strlen(n) + 1];
        uguulel[uguulel_too] = n;
        uguulel_too++;
    } else {
        cout << "Oguulel duursen." << endl;
    }
}

void sudlaach::search_uguulel(char n[]) {
    int i;

    for (i = 0; i < uguulel_too; i++) {
        if (strcmp(n, uguulel[i]) == 0) {
            cout << "\t" << uguulel[i] << endl;
        }
    }
}

```

```
}  
}
```

```
int main(){  
    char uguulel1[10] = "uguulel1";  
    char uguulel2[10] = "uguulel2";  
    char uguulel3[10];  
  
    sudlaach s;  
    s.read();  
    cout << "\n uguulel oruulna uu: ";  
    cin >> uguulel3;  
    s.add_uguulel(uguulel1);  
    s.add_uguulel(uguulel2);  
    s.add_uguulel(uguulel3);  
  
    s.print();  
  
    cout << "\n Haigaad oldson uguulel: \n";  
    s.search_uguulel(uguulel1);  
    s.search_uguulel(uguulel2);  
    s.search_uguulel(uguulel3);  
  
}
```