

С++ хэлний класс ба объект

Лаборатори 3

Э.Түвшин-Эрдэнэ(23B1NUM0869)

МТЭС, МКУТ-ын “Программ хангамж” хөтөлбөрийн оюутан, tuvshin.erdene25@gmail.com
2025 он

1. Оршил

Энэ лабораторийн ажлаар объект хандлагат программчлалын үндсэн ойлголт болох класс ба объект зэрэг ойлголтуудыг С++ хэл дээр судлан, тайлбарлав. Жишээ ажилчдын класс зарлан, энгийн хүний нөөцийн программ бичин кодыг үр дүнгийн хамт оруулав.

2. Зорилго

Класс ба объект, тэдгээрийн ялгаа, хэрэглээг С++ хэл дээр тайлбарлах, классын гишүүн өгөгдөл, функцийн холбоо хамаарал, өгөгдлийн битүүмжлэл, өгөгдлийн нууцлал ба гишүүн өгөгдөл, функцэд хандах тухай судлан оруулах.

3. Онолын судалгаа

3.1) Класс

Объект хандлагат программ нь классаас тогтоно. Класс бүр модуль болно. Классын шинж тэмдэг нь объектын шинж буюу төлөв байдал, үйл хөдлөл нь функц болно. Хамаарал нь класс хоорондын, объект хоорондын холбоог заана.[1]

Классыг объект үүсгэхдээ загвар болгон ашиглах ба түүний үйл хөдлөлийг функцээр тодорхойлно. Тэгвэл класс нь түүнээс үүссэн объектуудын дундаа хэрэглэх төлөв байдлын өгөгдөл, түүнийг боловсруулах функцийн нэгдэл, зохиомол өгөгдлийн төрөл юм.

Класс нь өөрийн гишүүн өгөгдөл, гишүүн функц бүхий хэрэглэгчийн тодорхойлсон өгөгдлийн төрөл.[5]

3.2) Объект

Объект гэж өөрийн нэртэй, тодорхой утга бүхий шинж тэмдэг, үйл хөдлөлийн төлөв бүхий, дүрслэн бодож болох аливаа юмыг хэлнэ.[1]

Харин объект хандлагат программчлалд объект нь тухайн классынхаа нэг жишээ, тохиолдол болно.

Объект бүр өөрийн гэсэн шинж тэмдэг, үйл хөдлөлтэй байх ба объектын шинжийг өгөгдөл(data), үйл хөдлөлийг нь *функц* (function) гэдэг.

Эндээс үзэхэд класс нь объектуудын ерөнхий хэв шинжийг тодорхойлсон загвар бол объект нь классын нэг тохиолдол юм.

3.3) Класс зарлах ба объект байгуулах

Объект байгуулахын өмнө түүний классыг зарлах хэрэгтэй ба классыг C++ хэл дээр дараах байдлаар зарлана:

```
Class <class_name> {
    <member datas>
    <member functions>
};
```

Класс зарлахдаа мөн хандалтын түвшнийг илэрхийлсэн public, private, protected зэрэг түлхүүр үгсийг ашиглаж болно.

Классын зарлалтаар ой бэлддэггүй, зөвхөн классын хувьсагч буюу объект үүсгэх логик загвар юм.

Объектыг байгуулахдаа классын нэрийг объектын нэртэй бичин байгуулна.

```
<Class_name> <object_name> ;
```

Хэрэв олон объект байгуулахаар бол:

```
<Class_name> <object_name1>, <object_name2>, <object_name3>;
```

Гишүүн өгөгдөл классын объект бүрд, гишүүн функц классын хувьд зөвхөн нэг удаа байгуулагдана. Нэг классын объектууд классын гишүүн функцийг дундаа хэрэглэнэ.

3.4) Гишүүн өгөгдөл, гишүүн функцийн хамаарал

Гишүүн функц классынхаа гишүүн өгөгдөл дээр боловсруулалт хийж программын бусад функцтэй мэдээлэл солилцоно.

Классын private гишүүн өгөгдөлтэй түүний функцээр нь дамжуулан хандах ба энэ тохиолдолд функцүүд нь класс болон тухайн дуудсан гадаад функц(main())-г.м) хооронд интерфейс болно.

3.5) Өгөгдлийн битүүмжлэл(data encapsulation)

Өгөгдлийг нэг нэгж болгон нэгтгэх арга. Код болон түүний боловсруулдаг өгөгдлийг нэгтгэнэ. Өөрөөр хэлбэл тухайн нэгжээс гаднах кодоор өгөгдөлд хандахаас сэргийлдэг.

- Өгөгдлийг битүүмжилснээр гишүүн өгөгдөлд өөр классаас хандаж болохгүй ба тухайн классын гишүүн функцээр л хандаж болно.
- Гишүүн өгөгдлүүдийг private, түүнд хандах функцүүдийг public хэлбэрээр зарлана.
- Setter болон getter функцүүдийг ашиглана.

Өгөгдлийг битүүмжилснээр дараах давуу талуудтай:

- Өгөгдлийн бүтэн бүрэн байдал хадгалагдана.
- Кодыг дахин ашиглах болон, шинэ шаардлагад нийцүүлэн өөрчлөлт хийхэд хялбар болно.
- Нэгжийн тест хийхэд хялбар.

Сул талууд нь:

- Зөвөөр хэрэглэхгүй бол код илүү төвөгтэй болно.
- Системийг ойлгоход хэцүү болно.
- Хэрэгжүүлэлтийн уян хатан байдлыг багасгана.[2]

3.6) Өгөгдлийн нууцлал (Data hiding)

Хэрэглэгч болон гадны этгээдээс өгөгдлийг нууна. Өгөгдөлд зөвшөөрөлгүйгээр өөрчлөлт хийхээс хамгаалах ба системийг илүү ойлгомжтой болгодог.[4]

Программын хамгийн чухал, эмзэг хэсэг болох өгөгдлийг хууль бус хандалтаас сэргийлэх арга зам. Үүний нэг жишээ нь банкны дансны үлдэгдлийг хэдий өөр программаас хардаг байж болох ч гэсэн өөрчлөлт оруулж болохгүй. Үүнээс сэргийлэх тулд дансны үлдэгдэлд зөвхөн тухайн классын функцээр ханддаг байна.[4]

Data hiding	Encapsulation
Өгөгдлийн аюулгүй байдалтай холбоотой.	Өгөгдлийг багцлан ганц нэгж болгоно гэж тодорхойлж болно.
Программын төвөгтэй хэсгийг нуух боломжтой.	Программын чухал, төвөгтэй мэдээллийг нууна.
Өгөгдлийн ашиглалтыг нуух болон хязгаарлахтай холбоотой.	Системийн төвөгтэй байдлыг нуухтай холбоотой.
Арга техникт тооцогддог.	Өгөгдлийг нуух ерөнхий процессийн дэд процесст тооцогддог.
Өгөгдөл үргэлж private, хандах боломжгүй байна.	Шаардлагаасаа хамаараад битүүмжлэгдсэн өгөгдөл private эсвэл public байж болно.

3.7) Гишүүн өгөгдөлд хандах

Классын гишүүн өгөгдөлд хандахдаа объектын нэр болон тухайн гишүүн өгөгдлийн нэрсийн дунд “.” шууд хандалтын оператор ашиглан хандана. Бүтцээс ялгаатай нь классын private өгөгдөл рүү шууд ингэж хандах боломжгүй байдаг.

<object_name>.<attribute_name>

3.8) Гишүүн функцийг дуудах

Объектын нэр болон гишүүн функцийн нэрсийг шууд хандалтын цэгээр зааглана.

<object_name>.<function_name>()

Энэ дүрмийг классын гишүүн функц дотор тухайн классынхаа өөр функцийг дуудаж байвал мөрдөхгүй.

4. Хэрэгжүүлэлт

4.1) Ажилтны класс зарлалт ба объект байгуулалт

Employee.h файл:

```
class Employee{
public:
    int id;
    char name[20];
    char position[10];
    float worked_hours;

    void start();
    void setInfo();
    void getInfo();

    float calcSalary();
    float bossSalary();
    bool inc_hour(float time);
};
```

Lab03.cpp файл:

```
#include<iostream>
#include<string.h>
#include"Employee.h"
using namespace std;

void Employee::start(){
    id = 0;
    strcpy(name,"");
    strcpy(position,"worker");
    worked_hours = 0;
```

```

}

void Employee::setInfo(){
    cout<<"enter id: ";cin>>id;
    cout<<"enter name: ";cin>>name;
    cout<<"enter position: ";cin>>position;
    cout<<"enter worked hours: ";cin>>worked_hours;
}

void Employee::getInfo(){
    cout<<"ID: "<<id<<endl;
    cout<<"name: "<<name<<endl;
    cout<<"position: "<<position<<endl;
    cout<<"worked hours: "<<worked_hours<<endl;
}

float Employee::calcSalary(){
    float hourly_pay;
    cout<<"enter hourly_pay";cin>>hourly_pay;
    if(strcmp(position, "boss")){
        return hourly_pay*worked_hours + bossSalary();//herew boss bol
        bonus nemj bodno.
    }
    return hourly_pay*worked_hours;
}

float Employee::bossSalary(){//boss iin bonus iig bodoj olno.
    float boss_hourly_pay;
    cout<<"enter hourly bonus of boss: ";cin>>boss_hourly_pay;
    return worked_hours*boss_hourly_pay;
}

bool Employee::inc_hour(float time){
    if(time<0||time>24){
        return 0;
    }
    worked_hours += time;
    return 1;
}

int main(){
    Employee e1;

```

```

    e1.start();
    e1.getInfo();

    cout<<endl;

    e1.setInfo();

    cout<<endl;

    e1.getInfo();
    return 0;
}

```

Дараа нь хэрэглэгч ашиглаж болохоор өөрчлөв:

```

#include<bits/stdc++.h>
#include"Employee.h"
using namespace std;

void Employee::start() {
    id = 0;
    strcpy(name, "");
    strcpy(position, "worker");
    worked_hours = 0;
}

void Employee::setInfo() {
    cout<<"enter id: ";cin>>id;
    cout<<"enter name: ";cin>>name;
    cout<<"enter position: ";cin>>position;
    cout<<"enter worked hours: ";cin>>worked_hours;
}

void Employee::getInfo() {
    cout<<"ID: "<<id<<endl;
    cout<<"name: "<<name<<endl;
    cout<<"position: "<<position<<endl;
    cout<<"worked hours: "<<worked_hours<<endl;
}

float Employee::calcSalary() {
    float hourly_pay;
}

```

```

        cout<<"enter hourly_pay: ";cin>>hourly_pay;
        if(strcmp(position, "boss")==0){
            return hourly_pay*worked_hours + bossSalary();//herew boss bol
bonus nemj bodno.
        }
        return hourly_pay*worked_hours;
    }
float Employee::bossSalary(){//boss iin bonus iig bodoj olno.
    float boss_hourly_pay;
    cout<<"enter hourly bonus of boss: ";cin>>boss_hourly_pay;
    return worked_hours*boss_hourly_pay;
}
bool Employee::inc_hour(float time){
    if(time<0||time>24){
        return 0;
    }
    worked_hours += time;
    return 1;
}

int main(){
    cout<<"Human resource program by Tuvshin-Erdene 2025"<<endl;
    vector <Employee> v;

    while(1){
        int choice;
        cout<<"1. list of employees 2. choose employee 3.new employee -1
to exit: ";
        cin>>choice;

        if(choice==1){
            for(int i = 0 ; i < v.size(); i++){
                cout<<i<<"."<<endl;
                v[i].getInfo();
            }
            cout<<"-----"<<endl;
            cout<<"total number of employees: "<<v.size()<<endl;
        }

        else if(choice==2){

```

```

        cout<<"Enter the id: ";
        int cID, num;
        cin>>cID;
        for(int i = 0 ; i < v.size(); i++){
            if(v[i].id==cID){
                num = i;break;
            }
        }
        v[num].getInfo();
        cout<<endl;
        int choice2;
        while(1){
            cout<<"1.set information 2.calculate salary 3.increase
work hour -1 to exit: ";
            cin>>choice2;
            if(choice2 == 1){
                v[num].setInfo();
            }
            else if(choice2==2){
                float salary = v[num].calcSalary();
                cout<<"salary = "<<salary<<endl;
            }
            else if(choice2==3){
                cout<<"enter hour: ";int time;cin>>time;
                v[num].inc_hour(time);
            }
            else{
                cout<<"exiting employee with id : "<<cID<<endl;
                break;
            }
        }

    }

    else if(choice == 3){
        Employee e1;
        e1.setInfo();
        v.push_back(e1);
        cout<<"Added successfully"<<endl;
    }

    else{

```



```

        break;
    }
}
return 0;
}

```

Энд C++ ийн vector өгөгдлийн төрлийг ашигласан ба ажилчдын жагсаалт гаргах, ажилчин сонгоод мэдээллийг нь өөрчлөх, шинэ ажилчин нэмэх боломжтой.

```

Human resource program by Tuvshin-Erdene 2025
1. list of employees 2. choose employee 3.new employee -1 to exit: 3
enter id: 1
enter name: tuvshin
enter position: boss
enter worked hours: 2
Added successfully
1. list of employees 2. choose employee 3.new employee -1 to exit: 2
Enter the id: 1
ID: 1
name: tuvshin
position: boss
worked hours: 2

1.set information 2.calculate salary 3.increase work hour -1 to exit: 2
enter hourly_pay: 12
enter hourly bonus of boss: 2
salary = 28
1.set information 2.calculate salary 3.increase work hour -1 to exit: 3
enter hour: 1
1.set information 2.calculate salary 3.increase work hour -1 to exit: -1
exiting employee with id : 1
1. list of employees 2. choose employee 3.new employee -1 to exit: -1

```

5.Дүгнэлт

Класс ба объект зэрэг ойлголтуудыг ашигласнаар бодит амьдрал дээрх зүйлсийг кодоор дүрслэхэд илүү дөхөм болж байна. Өгөгдөл болон өгөгдлийг боловсруулах функцийг нэгтгэн битүүмжилснээр программыг хийхэд илүү ойлгомжтой энгийн болсон. Хэдий битүүмжлэл болон өгөгдөл нууцлалтын ач холбогдлыг судалсан ч гэсэн хэрэгжүүлэхийн тулд хандалтын түвшний мэдлэг хэрэгтэй байна.

6.Ашигласан материал

1. “Объект хандлагат технологийн C++ програмчлал” Ж.Пүрэв 2008, Улаанбаатар

2. [Everhttps://www.simplilearn.com/tutorials/cpp-tutorial/data-hiding-in-cpp#:~:text=Data%20encapsulation%20focuses%20on%20wrapping,and%20how%20different%20objects%20behave,ything You Need to Know | Simplilearn](https://www.simplilearn.com/tutorials/cpp-tutorial/data-hiding-in-cpp#:~:text=Data%20encapsulation%20focuses%20on%20wrapping,and%20how%20different%20objects%20behave,ything%20You%20Need%20to%20Know%20|%20Simplilearn)
3. [GeeksforGhttps://www.geeksforgeeks.org/encapsulation-in-java/eeks](https://www.geeksforgeeks.org/encapsulation-in-java/eeks)
4. <https://www.tutorialspoint.com/difference-between-data-hiding-and-encapsulationBetween Data Hiding and Encapsulation>
5. <https://www.geeksforgeeks.org/data-hiding-in-python/>
6. <https://www.geeksforgeeks.org/object-oriented-programming-in-cpp>