

Класс Ба Түүний Хэрэглээ Лабораторын Ажил (Лаборатори №3)

Г.Энх-Амар

ХШУИС, МКУТ, Програмын хангамж, 18b1num0399@stud.num.edu.mn

1. ОРШИЛ/УДИРТГАЛ

Объект Хандлагат Програмчлал гол тулгуур болсон Класс хэрэглээтэй танилцаж, класс объектын ялгаа болон гишүүн функц, гишүүн өгөгдлийн талаар мэдэж авна.

2. ЗОРИЛГО

Оюутан багшийн өгсөн дараах 5 даалгаврыг биелүүлж, ажилтан классыг ойлгомжтой, уншууртай байдлаар хэрэгжүүлнэ. Үүнд:

1. Класс гэж юу болох, онцлог, үүргийн талаар бичих,
2. Класс болон C++ объектын ялгааг тайлбарлах,
3. Гишүүн функц, гишүүн өгөгдөл хоёр ямар хамааралтай байдаг вэ? Өгөгдлийн битүүмжлэл гэж юу вэ? гэсэн асуултанд хариулах,
4. Классын гишүүн өгөгдөл болон гишүүн функцэд хэрхэн хандах талаар бичих,
5. Ажилчин гэсэн класс тодорхойлно. Ажилчдын ажилласан цаг бүрийг өөрчилж цалинг тооцоолох жижиг програм бичих.

3. ОНОЛЫН СУДАЛГАА

3.1. Класс

C++ хэлнийг объект хандлагат технологийн гол хөдөлгүүр болгодог чухал шинж нь класс гэж нэрлэгддэг өгөгдөл хийсвэрлэл юм.

Өгөгдлийн бүтэц бол янз бүрийн төрлийн өгөгдлийн нэгдэл юм. Ийм бүтцэн өгөгдлийг зүйл бүрийн функцээр боловсруулна. Класс нь ийм функц болон өгөгдлийг хоорондоо холбоотой нэгэн цул битүүмжилсэн логик зүйл гэж үздэг. Иймд класс бол өгөгдөл ба функцийн нэгдэл, хэрэглэгчийн тодорхойлох зохиомол өгөгдлийн төрөл юм. Класс нь объектыг хялбар байгуулах үүрэгтэй.

Класс нь гишүүн өгөгдөл, гишүүн функцийн нэгдэл тул тодорхойлолт дотор нь өгөгдлийг тодорхойлж, функцийг гол төлөв зарлана.

Классын гол онцлог нь гишүүн өгөгдөлөө далд байдлаар зарлаж, зөвхөн гишүүн функцаараа дамжуулан уг өгөгдөлд хандах боломжийг олгодог. Ингэснээр өгөгдлийг санамсаргүй байдлаар өөрчлөхөөс сэргийлэхээс гадна, програмыг илүү эмх цэгцтэй уншууртай, ойлгомжтой болгож өгдөг.

Классыг зарлахдаа доорх байдлаар зарладаг. Классын гишүүдийн тодорхойлолт нь private/public/protected хэмээх гурван хэсэгтэй байж болно. Хэрвээ зааж өгөөгүй бол private байна.

```
Class class_name {  
...private/prptected/public:  
data_memebers;  
Private/publi:  
        member_functions;  
};
```

Private, public тусгай үгс классын гишүүдийн үйлчлэх хүрээг заана. Private гишүүд рүү зөвхөн тухайн класс дотроос, public гишүүд рүү хаанаас ч хандах боломжтой. Protected түвшин нь тухайн классынхаа хувьд private түвшинтэй адил боловч тэдгээр нь удамших байдлаараа хоорондоо ялгаатай.

3.2. Класс болон C++ объектын ялгаа

Хамгийн энгийнээр класс гэдэг нь ямар нэг зүйлийн модель бөгөөд уг моделийн дагуу үүсэж буй бүтээгдэхүүн зүйлийг объект гэж нэрлэнэ.

Жишээлбэл: амьтан гэх класс нь нэр, хөлний тоо, арьс гэсэн өгөгдлийг хадгалдаг байя.

```
class animal {  
    public:  
    char name[20];  
    int num_of_legs;  
    char skin[20];  
};
```

Энэхүү классыг ашиглан animal cat; cat.name = "cat", cat.num_of_legs = 4, cat.skin = "fur" гэсэн утга өгвөл. Cat гэсэн объект үүснэ.

Animal загвараа дахин ашиглаж animal duck; dove.name = "duck"; duck.num_of_legs = 2; duck.skin = "feather" гэсэн утга өгвөл Duck гэсэн объект үүснэ.

Эндээс бид класс нь объект үйлдвэрлэгч болохыг харж болно. Харин объект тухай классынхаа бүтээгдэхүүн юм.

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

Гишүүн өгөгдөл тухайн класстай холбоотой int, float, char гээд ямар нэгэн өгөгдөлийг хадгалдаг. Гишүүн функц нь гишүүн өгөгдөл дээр боловсруулалт хийж програмын бусад функцтэй мэдээлэл

солилцоно. Классын гишүүн функц ба гишүүн өгөгдлийг тодорхой түвшинд нээлттэй биш объектийн нэг хүрээнд оруулсныг өгөгдлийн битүүмжлэл гэнэ. Өөрөөр хэлбэл, объектын гишүүн өгөгдөлд зөвхөн гишүүн функцээр нь дамжуулан хандана гэсэн үг. Битүүмжилснээр өгөгдөлд хандах эрхийг хязгаарлах боломж олгоно. Иймд битүүмжлэл бол функц, өгөгдлийг тодорхой түвшинд нээлттэй биш объектын хүрээнд хамтад нь холбоотойгоор авч үздэг ойлголт юм.

3.4. Классын гишүүн өгөгдөл болон гишүүн функцэд хэрхэн хандах вэ?

Гишүүн өгөгдөл болон гишүүн функц нь public хэлбэрээр тодорхойлогдсон бол объектоор нь дамжуулан ханддаг. Хэрвээ private хэлбэрээр зарлагдсан гишүүн байвал тухайн гишүүн рүү public хэлбэрээр зарлагдсан функцаар дуудан хандах боломжтой.

4. ХЭРЭГЖҮҮЛЭЛТ

Бодлого : Ажилчин гэсэн класс тодорхойлно. Ажилчдын ажилласан цаг бүрийг өөрчилж цалинг тооцоолох жижиг програм бич.

1) Энэхүү программыг C++ хэл дээр бичих бөгөөд гараас утга авах, хэвлэх зорилгоор <iostream>, дэлгцэнд хэвлэхдээ хүснэгтэн байдлаар хэвлэхийн тулд <iomanip>, text харьцуулах, хуулах зорилгоор <cstring> гэсэн сангуудыг ашиглана.

Employee нэртэй класс зарлахдаа private хэлбэрээр гишүүн өгөгдөл болох ажилтны ID, нэр, албан тушаал, ажилсан цаг, цагийн хөлс зэрэг мэдээллийг авна. Харин public хэлбэрээр гишүүн функцуудаа тодорхойлох болно. Захиралын үндсэн цалинг кодын дээд хэсэгт #DEFINE DIRECTOR_BASE_SALARY 2000 өгсөн учир үндсэн цалин нь 2000 болно.

Гишүүн функцууд нь дараах үүрэгтэй:

- Void initialize() Тухайн ажилтны мэдээллийг 0, "" гэсэн гарааны утга өгнө.
- Void getdate() Тухайн ажилтанд мэдээлэл нэмнэ
- Void showdata() Тухайн ажилтны мэдээллийг хэвлэнэ
- Float calculate() Тухайн ажилтны цалинг тооцооно. Ажилсан цаг*Цагийн хөлс
- Float calculate_dir() Захиралын цалинг тооцно. Ажилтсан цаг*Цагийн хөлс + Үндсэн
- Bool add_hours() 0-24 утга авах ёстой бөгөөд тэр утгаа ажилсан цаг дээр нэмээд 1 гэсэн утга буцаана. Хэрвээ 0-24 хооронд байхгүй бол ажилсан цаг дээр нэмэхгүй бөгөөд 0 гэсэн утга буцаана.

Class хүснэгтийг доорх байдлаар зарлана.

```
- class employee
- {
-     private:
-         int id;
-         char name[20];
-         char position[10];
-         float worked_hours;
-         float hourly_wage;
```

```

-     public:
-         void initialize();
-         void getdata();
-         void showdata();
-         float calculate();
-         float calculate_dir(float);
-         bool add_hours(float);
-     };

```

Employee классын INITIALIZE() функц тухайн ажилтанд гарааны утгыг өгөх ёстой учир. Тоон төрлийн өгөгдлийг 0 болгож, Үсгэн төрлийн өгөгдлийг "" болгоно.

```

void employee::initialize()
{
    id = 0;                // Тоон төрлийн өгөгдөл
    strcpy(name, "");      // cstring сангийн функц бөгөөд "" утгыг
    strcpy(position, "");  // name, position-д өгнө
    worked_hours = 0;      //
    hourly_wage = 0;       //
}

```

Employee классын GETDATA() функц тухайн ажилтанд гараас утга өгөх болно. C++ хэлний cin, cout ашиглахад хялбар шийдэгдэнэ.

Employee классын CALCULATE() функц тухайн ажилтны цалинг тооцох болно. Хэрвээ уг ажилтан нь захирал CALCULATE_DIR() функцийг дуудна. Захиралын мөн эсэхийг шалгахын тулд cstring сангийн strcmp(position, "Director") == 0 гэсэн нөхцөлийг хангаж байгаа эсэхийг шалгана. Энэ нь position-г Director гэсэн өгөгдөлтэй ASCII кодоор жишиж ялгааг тоон утгаар харуулдаг. Ялгаа байхгүй бол 0 гэсэн утга буцаана. Тэгэхээр захирал мөн бол 0 гэсэн утга буцна.

```

float employee::calculate_dir(float dir_per_hour)
{
    float total = dir_per_hour*worked_hours + DIRECTOR_BASE_SALARY;
    return total;
}

float employee::calculate()
{
    if (strcmp(position, "Director") == 0 || strcmp(position, "director") == 0)
    {
        return calculate_dir(hourly_wage);
    }
}

```

```
    return worked_hours*hourly_wage;
}
```

Employee классын ADD_HOURS() функц 0-24 гэсэн утга авч байгаа эсэхийг шалгаж, тийм байвал ажилсан цаг дээр тухайн утгыг нэмээд 1 гэсэн утга буцаадаг, эсрэг тохиолдолд 0 гэсэн утга буцаадаг.

```
bool employee::add_hours(float adding_hours)
{
    if (adding_hours >= 0 && adding_hours <= 24)
    {
        worked_hours += adding_hours;
        return 1;
    }
    return 0;
}
```

Employee классын showdate() функц нь тухайн ажилтны мэдээллийг харуулдаг. Энэ хэсгийг энгийн cout-аар хийв.

```
void employee::showdata()
{
    int grosspay = calculate();
    cout << setw(15) << id;           // Ажилтны дугаар
    cout << setw(20) << name;         // Нэр
    cout << setw(20) << position;     // Албан тушаал
    cout << setw(15) << worked_hours;  // Ажилсан цаг
    cout << setw(18) << hourly_wage;  // Цагийн хөлс
    cout << setw(10) << grosspay;     // Нийт цалин
    cout << endl;
}
```

2)

Програм маань анх ажилахдаа хэдэн ажилтан байгааг асууж ажилтан классаар үүссэн объект болгондоо гарааны утга оноосон.

"D:\EDUCATION\NUM\6th semester\oop\lab3\main.exe"

Enter number of employees: 3

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

1

Employee ID	Employee Name	Position	Hours	Hourly Wage	Gross
0			0	0	0
0			0	0	0
0			0	0	0

Нийт 3 ажилтныгаа бүх мэдээллийг харахад гарааны утга өгсөн байна.

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

3

Enter the row number: 1

Өгөгдөл оруулах функцийг дуудав.

1-р мөрт өгөгдөл оруулъя.

Enter the Employee ID: 100

Enter the Employee Name: Enkh-Amar

Enter the Position: Director

Enter the Worked Hours: 37

Enter the Hourly Wage: 11.5

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

3

Enter the row number: 2

Өгөгдөл оруулах функцийг дуудав.

2-р мөрт өгөгдөл оруулъя.

Enter the Employee ID: 101

Enter the Employee Name: Ganbat

Enter the Position: Manager

Enter the Worked Hours: 40.5

Enter the Hourly Wage: 10.5

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

1

Оруулсан өгөгдөлөө харцгаая.

Employee ID	Employee Name	Position	Hours	Hourly Wage	Gross
100	Enkh-Amar	Director	37	11.5	2425
101	Ganbat	Manager	40.5	10.5	425
0			0	0	0

Объектонд гарааны утгах оноох функц алдаагүй ажиллаж байна.

"D:\EDUCATION\NUM\6th semester\oop\lab3\main.exe"

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

2

Enter the row number: 1

Succeed!

АМЖИЛТТАЙ болсон байна.

Объектодоо гарааны утга олгоё

1-р мөрт байгаа объектыг сонгоё

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit

1

1-р мөр өөрчлөгдсөн эсэхийг шалгахын тулд бүх объектуудыг хэвлэе

Employee ID	Employee Name	Position	Hours	Hourly Wage	Gross
0			0	0	0
101	Ganbat	Manager	40.5	10.5	425
0			0	0	0

Амжилттай

Программын ажилсан цагийг нэмэх хэсэг асуудалгүй ажиллаж байна.

```

"D:\EDUCATION\NUM\6th semester\oop\lab3\main.exe"
1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit
3
Enter the row number: 1

Enter the Employee ID: 100
Enter the Employee Name: Huyagbaatar
Enter the Position: director
Enter the Worked Hours: 22
Enter the Hourly Wage: 11.5

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit
1
-----
Employee ID      Employee Name      Position      Hours      Hourly Wage      Gross
-----
100              Huyagbaatar        director        22          11.5             2253
101              Ganbat             Manager         40.5         10.5             425
0

```

1-р мөрт байгаа
объектод 100 ажилтны
дугаартай, захирал
албан тушаалатай
Хуягбаатар нэртэй, нийт
22 цаг ажилсан, цагийн
үнэлгээ нь 11.5 байх утга
оноов.

```

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit
4
Enter the row number: 2
Enter the amount of hours: 20
Succeed!

1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit
1
-----
Employee ID      Employee Name      Position      Hours      Hourly Wage      Gross
-----
100              Huyagbaatar        director        22          11.5             2253
101              Ganbat             Manager         60.5         10.5             635
0

```

2-р мөрт байгаа объектын
ажилсан цагийг 20-р
нэмэгдүүлэхэд 60,5
болсон байна.

5. ДҮГНЭЛТ

Объект хандлагат технологийн чухал хэсэг болох КЛАСС-ын тусламжтай бид ямархуу объект үүсгэх вэ гэдгийг загварчлах боломжтой. Класс өгөгдлийн төрөл, харин объект хувьсагчтай адил үйлчилж байна. Мөн объектын private гишүүдүүд гаднаас хандах боломжгүй учир өгөгдлийг санамсаргүй байдлаар өөрчлөхөөс сэргийлдэг.

6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.
2. <http://mmunkherdene.blogspot.com/2013/06/m-objekt-handlagat-pragramchlal.html>
3. Iomanip setw, <https://www.cplusplus.com/reference/iomanip/setw/>

7. ХАСВРАЛТ

```

#include <iostream>
#include <iomanip>
#include <cstring>

#define DIRECTOR_BASE_SALARY 2000

using namespace std;

class employee
{

```

```

private:
    int id;
    char name[20];
    char position[10];
    float worked_hours;
    float hourly_wage;
public:
    void initialize();
    void getdata();
    void showdata();
    float calculate();
    float calculate_dir(float);
    bool add_hours(float);
};

void employee::initialize()
{
    id = 0;
    strcpy(name, "");
    strcpy(position, "");
    worked_hours = 0;
    hourly_wage = 0;
}

void employee::getdata()
{
    cout << "\nEnter the Employee ID: ";
    cin >> id;

    cout << "Enter the Employee Name: ";
    cin >> name;

    cout << "Enter the Position: ";
    cin >> position;

    cout << "Enter the Worked Hours: ";
    cin >> worked_hours;

    cout << "Enter the Hourly Wage: ";
    cin >> hourly_wage;
}

float employee::calculate_dir(float dir_per_hour)
{
    float total = dir_per_hour*worked_hours + DIRECTOR_BASE_SALARY;
}

```



```

        return total;
    }

float employee::calculate()
{
    if (strcmp(position, "Director") == 0 || strcmp(position, "director") == 0)
    {
        return calculate_dir(hourly_wage);
    }
    return worked_hours*hourly_wage;
}

bool employee::add_hours(float adding_hours)
{
    if (adding_hours >= 0 && adding_hours <= 24)
    {
        worked_hours += adding_hours;
        return 1;
    }
    return 0;
}

void employee::showdata()
{
    int grosspay = calculate();
    cout << setw(15) << id;
    cout << setw(20) << name;
    cout << setw(20) << position;
    cout << setw(15) << worked_hours;
    cout << setw(18) << hourly_wage;
    cout << setw(10) << grosspay;
    cout << endl;
}

void heading()
{
    cout << endl;
    for (int k = 0; k < 99; k++)
        cout << "-";
    cout << endl;
    cout << setw(15) << "Employee ID";
    cout << setw(20) << "Employee Name";
    cout << setw(20) << "Position";
    cout << setw(15) << "Hours";
    cout << setw(18) << "Hourly Wage";
}

```

```

        cout << setw(10) << "Gross";
        cout << endl;

        for (int k = 0; k < 99; k++)
            cout << "-";
        cout << endl;
    }

int main()
{
    int num_of_employees;
    cout << "Enter number of employees: ";
    cin >> num_of_employees;
    employee emp[num_of_employees];

    for (int k = 0; k < num_of_employees; k++)
        emp[k].initialize();

    int choose, row_num;
    float hours_to_add;
    while (1)
    {
        cout << "\n1: Show all, 2: Initialize, 3: Insert Data, 4: Add worked hours, 5: Exit\n";
        cin >> choose;

        switch (choose)
        {
            case 1:
                heading();
                for (int k = 0; k < num_of_employees; k++)
                    emp[k].showdata();
                break;
            case 2:
                cout << "Enter the row number: ";
                cin >> row_num;
                emp[row_num-1].initialize();
                cout << "Succeed!\n";
                break;
            case 3:
                cout << "Enter the row number: ";
                cin >> row_num;
                emp[row_num-1].getdata();
                break;

```

```
        case 4:
            cout << "Enter the row number: ";
            cin >> row_num;
            cout << "Enter the amount of hours: ";
            cin >> hours_to_add;
            if (emp[row_num-1].add_hours(hours_to_add))
                cout << "Succeed!\n";
            else cout << "INSERT VALUE BETWEEN 0 AND 24";
            break;
        default:
            exit(0);
    }
}

return 0;
}
```