

Хийсвэр Класс, Функц Тодорхойлох Лабораторын Ажил

(Лаборатори №7)

ХШУИС, МКУТ

Компьютерийн ухааны хөтөлбөр

Г.Мөнх-Оргил 18b1num2048

1. ОРШИЛ/УДИРТГАЛ

Объект хандлагат технологийн үндсэн суурь ойлголт болох удамшил, удамшлын модыг ашиглан хийсвэр функцуудын тусламжтай хоёр хэмжээст геометр дүрсүүдийг байгуулж, удамших объектууд дээр ажиллах.

2. ЗОРИЛГО

Дүрс эх класс болон түүний дэд классууд болох хоёр хэмжээст класс , тойрог, квадрат, ижил талт гурвалжин классуудыг, мөн хийсвэр класстай ажиллана. Үүний тулд дараах зорилтуудыг тавьж ажиллана.

1. Хийсвэр функц гэж юу вэ ?
2. Жинхэнэ хийсвэр функц гэж юу вэ?
3. Функц дахин програмчиллах гэж юу болох.
4. Хийсвэр класс гэж юу вэ ? Хэрхэн класс байгуулдаг вэ ?
5. Удамшил ба байгуулагч функц.
6. Удамшил ба устгагч функц.

3. ОНОЛЫН СУДАЛГАА

3.1. Хийсвэр функц гэж юу вэ?

Хийсвэр функц гэдэг нь эх класс дотор зарлагдсан, удамшсан буюу хүүхэд классаар дахин тодорхойлогдсон гишүүн функц юм. Эх классын хаягаар нь дамжуулж, хүүхэд классын объектын функцыг зөв дуудах боломжийг олгодог.

Хийсвэр функцийг зарлахдаа зөвхөн эх классын гишүүн функцийн өмнө virtual гэсэн түлхүүр үгийг бичих замаар зарлана.

Жишээ нь:

```

class base_class
{
    public:
        virtual void show(){
            cout << "show of base\n";
        }
};
class derived_One : public base_class
{
    public:
        void show(){
            cout << "show of one\n";
        }
};
class derived_Two : public base_class
{
    public:
        void show(){
            cout << "show of two\n";
        }
};

```

3.2. Жинхэнэ хийсвэр функц гэж юу вэ?

Функцийн тодорхойлолтгүй эх бие нь NULL утгатай байдаг функцийг жинхэнэ хийсвэр функц гэнэ. Хийсвэр арга нь их биегүй аргыг хэлнэ. Эх классд тодорхойлогдсон хийсвэр аргын хэрэгжилтийг хүү классууд нь өөр өөрийнхөөрөө хэрэгжүүлдэг. Жинхэнэ хийсвэр функцийг зарлахдаа:

```

virtual void MyVirtualMethod() = 0;

```

3.3. Функц дахин програмчлах

Эх классын функцийг хүүхэд классд бичдэс нь ижилээр дахин бичихийг/тодорхойлохыг функц дахин тодорхойдох гэнэ. (Overriding) Эх классд дахин програмчлах функцын өмнө с++ програмчлалын хэлэнд virtual гэсэн түлхүүр үгийг ашигладаг. Удамшсан классдаа тааруулан дахин тодорхойлж ашигладаг.

3.4 Хийсвэр класс

Жинхэнэ хийсвэр функцийг агуулсан классыг хийсвэр буюу abstract class гэнэ. Хийсвэр классаас объект үүсдэггүй. Харин түүнээс удамшсан классд хийсвэр функцийг нь дахин програмчилсны дараа удамшсан классын объектыг байгуулж болно.

Хийсвэр класс нь дараах онцлогтой. Үүнд:

- Бүрэн бус класс бөгөөд түүнээс заавал дэд класс удамшина.

- Дор хаяж нэг буюу түүнээс олон хийсвэр арга агуулсан байх.
- Түүний объектыг үүсгэх боложгүй.
- Түүний хүү класс нь түүний хийсвэр аргуудыг биелүүлнэ эсвэл түүний хүү класс нь өөрөө хийсвэрээр зарлагдах гэсэн онцлог шинжүүдтэй.

Хийсвэр классын энгийн жишээ нь:

```
class TwoDimen:public shape{
public:
    TwoDimen();
    TwoDimen(string col, int a, int b);
    virtual float findS() = 0;
    |
};
```

3.5. Удамшил ба байгуулагч функц

Хүүхэд классын объектыг байгуулах үед эх классын анхдагч байгуулагч эхлээд дуудагддаг ба үүний дараа хүүхэд классын байгуулагч дуудагдана. Хүүхэд классын параметртэй байгуулагчийг тодорхойлохдоо аргуудыг эх класс руу дамжуулна. Эх классаас удамшиж ирсэн гишүүн өгөгдлийг эх классын гишүүн функцээр нь утга оноолгуулах хэрэгтэй болдог.

```
TwoDimen::TwoDimen(string col, int x, int y):shape(col,x,y)
```

3.6. Удамшилд устгагч функц

Удамших классын объект устахад эхлээд удамших классын устгагч, дараагаар нь эх классын устгагч тус тус дуудагдана. Нэмэлт ажил хийлгэхгүй бол устгагч функцийг програмд тодорхойлох шаардлага байдаггүй ба ийм тохиолдолд анхдагч устгагчийг систем өөрөө үүсгэдэг.

Харин их классын устгагчийг virtual тодотгож өгсөнөөр устгах объектонд харгалзах устгагч функц нь бас дуудагдах боломжтой болдог.

4. ХЭРЭГЖҮҮЛЭЛТ

Дүрсний периметр болон талбайг олох функцүүд нь хийсвэр функц бөгөөд функцийг дуудхад гараас ямар нэгэн утга авахгүй шууд хариуг тооцоолон буцаана.

```
public:
    int radius;
    float findS()
    {
        return 3.14*radius*radius;
    }
    float perimeter()
    {
        return radius*3.14*2;
    }
```

```

class shape
{
    protected:
        int x;
        int y;
        char name;
    public:
        virtual float perimeter()= 0;
};

class twoDimen:public shape
{
    protected:
        int coordinate;
    public:
        virtual float findS() = 0;
        void showdata();
};

```

Тус бүр 10 ширхэг тойрог, гурвалжин, тэгш өнцөгтүүдийг талбайг олж, эрэмбэлсэн

```

void showdata(){
    int i;
    cout<<"\n~~~~~Circle~~~~~\n";
    circle toirog[11];
    cout<<"eremblegdeegvi: "<<endl;
    for(i = 1; i<=10; i++){
        cout <<"toirgiin radius:";
        cin >>toirog[i].radius;
        cout << "toirgiin perimeter:" << toirog[i].perimeter() << endl;
        cout <<"toirgiin talbai:"<< toirog[i].findS()<< endl;
    }
    for(i = 1; i <=10; i++){
        circle key = toirog[i];
        int j = i-1;
        while( j >= 0 && toirog[j].findS() < key.findS() ){
            toirog[j+1] = toirog[j];
            j = j-1;
        }
        toirog[j+1] = key;
    }
    cout<<"\nEremblegdsen talbai: "<<endl;
    for(i = 1; i<=10; i++)
        cout<<toirog[i].findS()<<endl;
}

```

5. ДҮГНЭЛТ

Дээрх ажлын хүрээнд объект хандлагат технологийн үндсэн ойлголтуудын нэг болох удамшлын шинж чанарыг ашиглан тойрог, гурвалжин, квадрат гэх мэт дүрсүүдийн классуусыг хийсвэрээр тодорхойлж өгөгдсөн шаардлагын дагуу үүгэсэн дүрсүүдийнхаа объектуудыг талбайгаар нь эрэмбэлэх ажил хийж гүйцэтгэлээ. Классуудыг байгуулж байх явцад эх классаас өөр онцлог шинжээ хадгалж үлдэхийн тулд функц болон классыг хийсвэрээр зарлаж өгөх нь зөв шийдэл болж өгсөн мөн удамшил ашигласан учир мөр бичлэг хэмнэх, алдаа гарахаас сэргийлэх, цаг хугацаа хэмнэх гэх мэт давуу талууд гарлаа.

6. Ашигласан материал

1. 1. Объект хандлагат технологийн C++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар
2. Virtual, <https://www.geeksforgeeks.org/pure-virtual-functions-and-abstract-classes/>

7. ХАВСРАЛТ

Кодыг ZIP файлын хавсралтанд оруулсан.