

*Лабораторийн Ажил  
(Лаборатори №6)  
Ц.Намсрайжамц*

*МУИС, ХШУИС, МКУТ, Программ хангамжийн хөтөлбөрийн 3-р түвшний оюутан  
18B1NUM1671@stud.num.edu.mn*

1. **ОРШИЛ/УДИРТГАЛ**-Энэ лабораторийн ажлаар удамшил, удамшилийн төрөл горимын талаар судлах мөн удамшлын хэрэгжүүлэлтийн C++ болон Java хэл дээрх ялгааг мэдэх

2. **ЗОРИЛГО**-Удамшлын талаарх мэдлэгээ ашиглан shape гэдэг классаас 2DShape классыг 2DShape -ээс Circle, Square, Triangle классуудыг удамшуулан удамшлыг хэрэгжүүлэх. Circle, Square, Triangle классуудын шинж өгөгдсөн бөгөөд тэдгээрээс үндэслэн эх классын шинжийг ерөнхийлөн гаргана.

3. **ОНОЛЫН СУДАЛГАА**

1. Удамшил гэж юу вэ? (тодорхойлолт, C++/java хэл дээр хэрхэн хэрэгжүүлдэг талаар бичнэ)

Өмнө байсан классын шинж тэмдэг, үйл хөдлөлийг бүтнээр эсвэл заримыг нь өөртөө хадгалан авч дээр нь өөрийн онцлог шинжийг нэмж, шинээр класс үүсэх үйл явцыг удамшил гэнэ.

**C++хэлэнд**-Өөрөөс нь шинэ класс үүсч буй классыг- үндсэн класс/дээд класс/эх класс гэнэ. Харин шинээр үүсч буйг нь-удамших класс/удамшсан класс(derived)/дэд класс(subclass)/охин класс гэж нэрлэдэг. /Си++ програмчлалын хэлэнд удамшилыг тодорхойлон бичихдээ эхлээд удамшин шинээр бий болж буй классын нэрийг/child class/ :/шинэ ба эх классын хамаарал заагч түүний дараа удамшуулсан классын нэрийг /parent class / тус тус бичдэг.

Бичиглэл — `class _2dshape : public shape{}`

**Java хэлэнд**- Өмнө нь оршин байсан классыг-эцэг класс/parent/ гэх ба түүнээс удамшин үүссэн классыг-хүү класс/child/ гэж нэрлэдэг.1 /Жава дээр удамшилыг тодорхойлон бичихдээ extends түлхүүр үгийн ар талд удамшуулсан классаа/parent class/ харин өмнө нь удамшин шинээр бий болж буй/child class/ классын нэрийг тус тус бичнэ.

Бичиглэл - `class 2dshape extends shape { }`

2. Удамшлын горим. public, private, protected горимын талаар тайлбарлаж жишээгээр батална.

**Public**- Эх класс public горимоор удамших тохиолдолд түүний өгөгдөлүүд хандалтын түвшингээ өөрчлөхгүй.

```

class ehClass
{
private:
    int a;

protected:
    int b;

public:
    int c;
    ehClass()
    {
        a = 5;
        b = 4;
        c = 3;
    };
    void getdata()
    {
        cout << a << b << c;
    }
};

class huuhedClass : public ehClass
{
public:
    void childGetData()
    {
        cout << a << b << c; //ene uildei ajillahgui buguud uchir ni huuhedclass-aas eh classiin private tuvshintei a
ugugdulruu handaj chadhgui
    };
};

int main()
{
    huuhedClass *a;
    a = new huuhedClass;
    a->childGetData();
}

```

**Private-** Эх классын public болон protected хандалтын түвшинтэй өгөгдөлүүд private түвшинтэй болно.

```

class huuhedClass : private ehClass
{
public:
    void childGetData(){};
};

int main()
{
    huuhedClass *a;
    a = new huuhedClass;
    a->childGetData();
}

```

```
a->getdata; //private gorimoor udamshsan tul public tuvshintei baisn getdata() functioniig duudan ajilluulj болохgui
}
```

**Protected-** Эх классын гишүүн өгөгдөл болон гишүүн функц хүүхэд класст protected болж өвлөгдөнө

### 3. Удамшлын давуу талуудыг тоочин бичиж бодит жишээн дээр тайлбарла.

Удамшил хэрэглэхийн давуу тал

- Дахин ашиглах боломжтой болно-Эх класстай холбоотой зүйл хийх болгонд дахин ашиглах боломжтой
- Найдвартай буюу алдаа бага гарна-өмнө нь бичигдсэн зүйлийг ашиглаж байгаа тул найдвартай
- Бичиглэл хэмнэнэ(өртөг)-нэг эх классаа ашиглан хэдэн ч удамшил үүсгэх боломжтой учир өртөг хэмнэнэ
- Цаг хугацаа хэмнэнэ-Заавал ахин бичих шаардлагагүйгээр өмнө нь хэрэглэсэн эх классаа хэрэглэх

### 4. Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бич.

- **Дан удамшил**-Дан удамшил буюу энгийн удамшил. Энэ нь удамших класс зөвхөн нэг эх класстай байна.  
A->B
- **Нийлмэл удамшил**-Энэ төрлийн удамшил нь 2с цөөнгүй эх класстай байна.  
A&B->C
- **Олон түвшинт удамшил** Удамших класс нь өөрөө мөн удамших классаас үүссэн байна.  
A->B B->C C->D
- **Шаталсан удамшил**-Энгийн удамшилтай төстэй. Гэхдээ 1 классаас 2-оос цөөнгүй класс удамшсан байна.  
A->B A->C A->D
- **Холимог удамшил**-Шаталсан болон нийлмэл удамшлын нэгдэл гэж ойлгож болно.  
A->B B->C B->D B->E

#### 4. ХЭРЭГЖҮҮЛЭЛТ

```
*****
1.Circle    2.Square  3.Triangle  4.Exit
*****

2
Taliin urtaa oruulna uu = 3
Zuun deed oroin koordinat oruulna uu !
a_x = 1
a_y = -2

*****
1.Talbai Oloh    2. Perimeter oloh    3. Dursiin medeelel 4.exit
*****
1
Talbai = 9

*****
1.Talbai Oloh    2. Perimeter oloh    3. Dursiin medeelel 4.exit
*****
3
Ene dursiin ner = Square
4 untsgiin coordinate : (1,-2) (4,4) (4,1) (1,-5)

*****
1.Talbai Oloh    2. Perimeter oloh    3. Dursiin medeelel 4.exit
*****
3
Ene dursiin ner = Square
4 untsgiin coordinate : (1,-2) (4,4) (4,1) (1,-5)

*****
1.Talbai Oloh    2. Perimeter oloh    3. Dursiin medeelel 4.exit
*****
4
```

```
class shape
{
protected:
    char *name;

public:
    shape();
    ~shape();
};

shape::shape()
{
    name = new char[10];
    strcpy(name, "");
}

shape::~~shape()
{
    delete name;
}

//2dshape class -----
class _2dshape : public shape
```

```
{
protected:
    int Perimeter, Taliin_urt;
    float Talbai;

public:
    int talbaiOloh();
    _2dshape();
    ~_2dshape();
};
```

## 5. ДҮГНЭЛТ

Объект хандалгат програмчлалын хэлийг бусад хэлнээс ялгаж буй гол зүйлсийн нэг бол удамшил юм. Удамшилын үүрэг шалтгааныг, төрөл, давуу талыг энэ лаборатарын ажлаар судлан мэдэж авсан.

## 6. АШИГЛАСАН МАТЕРИАЛ

Ж.Пүрэв Объект хандлагат технологийн C++ програмчлал