

УДАМШИЛИЙН ТАЛААРХ ОЙЛГОЛТ (ЛАБОРАТОРИ №6)

Э.Уранчимэг

ХШИУС, МКУТ, Программ хангамжийн III түвшний оюутан,
21B1NUM0609@stud.num.edu.mn

1. ОРШИЛ

Энэ тайлангийн онолын судалгаа хэсэгт удамшил түүний тодорхойлолт c++, java хэлүүд дээр хэрхэн хэрэгжүүлдэг талаар мөн удамшлын горим protected-ийн талаар жишээгээр тайлбарлан харуулсан. Удамшлын давуу талыг тоочин бичиж, жишээн дээр үзүүлсэн ба удамшлын төрөл тус бүрийг дэлгэрүүлэн судалж, тайлбарласан. Хэрэгжүүлэлт хэсэгт shape, 2DShape классуудын шинжийг ерөнхийллийн аргаар зохиомжилж удамшуулсан хэрэгжүүлэлтийг харуулсан ба тойрог, квадрат, зөв гурвалжин зэрэг классуудыг тодорхойлон, хийгдсэн үйлдлүүдийн кодыг үзэх боломжтой. Хавсралт хэсгээс дэлгэрэнгүй кодыг унших боломжтой.

2. ЗОРИЛГО

Энэ дасгалыг хэрэгжүүлсний үр дүнд бид удамшил болон түүнтэй холбоотой ойлголтуудыг дасгалаар хэрэгжүүлж, удамшлын горимуудын түлхүүр үгийг ашиглан эх классын гишүүд рүү хандаж сурна. Үүний тулд дараах зорилтуудыг тавьж ажилласан:

1. Удамшил түүний талаар онолын судалгаа хийх,
2. Shape классд дүрсний нийтлэг шинжийг тодорхойлох,
3. Shape классаас 2DShape классыг удамшуулах,
4. 2DShape классаас circle, triangle, square классуудыг удамшуулах,

3. ОНОЛЫН СУДАЛГАА

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

Тодорхой нэг классын шинж чанарууд, үйлдлүүдийг өөр нэг класс дотор дахин тодорхойлолгүй шууд авч ашиглах боломжийг удамшил гэж нэрлэдэг бөгөөд энэ нь эцэг класс, охин класс гэсэн ойлголтыг бий болгодог. Удамшлаар үүсэх класс өмнөх классынхаа бүх эсвэл зарим шинжийг өвлөн авах ба үүнийг байгаа классаас шинээр класс үүсгэх удамшлын механизм гэнэ.

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

Удамшлын горим гэдэг нь эх классын гишүүн өгөгдөл, функцуудын хандалтын түвшин нь хүүхэд классд хэрхэн өөрчлөгдөхийг тодорхойлдог ойлголт юм. Класс public, private, protected горимоор удамшиж болно.

1. *Public* - Эх класс public горимоор удамших тохиолдолд хүүхэд классаас эх классын гишүүдэд ямар нэгэн хандалтын горимын нэмэлт өөрчлөлтгүй шууд байгаа горимоор нь хандаж болдог.
2. *Private* - Энэхүү горимоор удамшуулсан хүүхэд класс эх класс руугаа шууд хандах боломжгүй байдаг бөгөөд бүх гишүүд нь хандах боломжгүй private хэлбэртэй болж удамшдаг.
3. *Protected* - Protected горимоор удамших тохиолдолд эх класст байгаа гишүүн өгөгдөл гишүүн функц protected болж удамшина. Энэхүү горим нь харьцангуй уян хатан буюу гаднаас хандах шууд боломжгүй хэдий ч хүүхэд классаас хандах боломжийг бий болгодог. Public болон Protected гишүүдэд шууд хандах боломжтой.
 - Эх классын private гишүүд өвлөгдөхгүй тул хүүхэд классын гишүүд эх классынхаа private гишүүд рүү хандаж чадахгүй.
 - Хүүхэд классын функц ба объект эх классынхаа public гишүүдэд шууд хандаж болно
 - Эх классын protected гишүүд хүүхэд классынхаа protected гишүүд болох тул хүүхэд классын функцээс хандаж болно.

3.3 Удамшлын давуу талуудыг тоочин бичиж бодит жишээн дээр тайлбарла.

Давуу талууд:

1. Удамшлыг ашигласнаар ерөнхий бүхий классыг шинэ класс үүсгэн өмнөх классын мэдээллийг өөрчлөлгүйгээр дахин ашиглах боломжийг бий болгодог.
2. Тухайн эх классаас үүдэлтэй олон хүүхэд класс үүсгэх бол удамшлыг ашиглан илүү бага кодын тусламжтайгаар бичих мөн ямар нэгэн эх классаас хамаарсан алдаа харьцангуй багасна.
3. Илүү бага зардлаар буюу энэхүү классаа ашиглан бүтээмжийг ихэсгэх боломжтой.

3.4 Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бич.

Удамшлын 5 төрөл байдаг. Үүнд :

1. *Нэг-нэг буюу дан энгийн удамшил* - Удамших класс зөвхөн нэг эх класстай байна.
2. *Нийлмэл удамшил* - Удамших класс нь 2-оос цөөнгүй эх класстай байж болно.
3. *Олон түвшинт удамшил* - Эх классаас удамшсан дэд класс нь дахин эх класс болж удамшуулана.
4. *Шаталсан удамшил* - Нэг эх классаас 2-оос цөөнгүй класс удамшихыг шаталсан удамшил гэнэ.
5. *Холимог удамшил* - Нийлмэл удамшил болон шаталсан удамшил хоёрын нийлэмж удамшил.Өөрөөр хэлбэл, нэг эх классаас 2-оос цөөнгүй класс удамшина. Хэд хэдэн удамшисан классаас 1 класс удамшихыг хэлнэ.

4. ХЭРЭГЖҮҮЛЭЛТ

Shape классаас 2DShape гэдэг класс удамшуулж, 2DShape классаас Circle, Square, Triangle классуудыг удамшуулсан ба Shape, 2DShape классуудын шинж, үйлдлийг ерөнхийллийн аргаар зохиомжилж удамшуулсан.

Дүрс класст нэрийг нь, 2D дүрс класст координат болон талыг нь шинж гэж үзэн зохиомжилсон бөгөөд 2D дүрс класст талбай олох, периметр олох гишүүн функцууд болон байгуулагч функцууд бичсэн. Мөн хүүхэд классуудад нь функц дахин тодорхойлсон.

Тойрог:

```
class circle : public twoDshape {
public:
    float findArea(){
        return 3.14 * l * l;
    }
    float findPerimeter(){
        return 2 * 3.14 * l;
    }

    circle(const char *name, int x, int y, int l) : twoDshape(name, x, y,
l) {
        cout << "Toirog(para) usev" << endl;
    }

    ~circle() {
        cout << "Toirog ustav" << endl;
    }

    void print() {
```

```

        cout << "Ner: " << get_name() << endl;
        cout << "Toirgiin tuv: " << x << " " << y << endl;
        cout << "Radius: " << l << endl;
        cout << "Talbai: " << findArea() << endl;
        cout << "Toirgiin urt " << findPerimeter() << endl;
    }
};

```

Квадрат:

```

class square : public twoDshape {
private:
    int x1, x2, x3, y1, y2, y3;

public:
    float findPerimeter(){
        return 4 * l;
    }

    square(const char *name, int x, int y, int l) : twoDshape(name, x, y,
l) {
        x1 = x + l;
        y1 = y;
        x2 = x + l;
        y2 = y + l;
        x3 = x;
        y3 = y + l;
        cout << "Kvadrat(para) uusev" << endl;
    }

    ~square() {
        cout << "Kvadrat ustav" << endl;
    }

    void print() {
        cout << "Ner: " << get_name() << endl;
        cout << "Koordinatuud: " << endl;
        cout << "a(" << x << ", " << y << ") ";
        cout << "b(" << x1 << ", " << y1 << ") " << endl;
        cout << "c(" << x2 << ", " << y2 << ") ";
        cout << "d(" << x3 << ", " << y3 << ") " << endl;
    }
};

```

```

        cout << "Urt: " << l << " " << endl;
        cout << "Talbai: " << findArea() << endl;
        cout << "Perimeter: " << findPerimeter() << endl << endl;
    }
};

```

Зөв гурвалжин:

```

class triangle : public twoDshape {
private:
    int x1, x2, y1, y2;

public:
    float findArea(){
        return sqrt(3)/4 *l*l;
    }
    float findPerimeter(){
        return 3 * l;
    }

    triangle(const char *ner, int x, int y, int l) : twoDshape(ner, x, y,
l) {
        y1 = l * cos(30 * 3.14 / 180) + y;
        y2 = l * cos(30 * 3.14 / 180) + y;
        x1 = l * sin(30 * 3.14 / 180) + x;
        x2 = x - l * sin(30 * 3.14 / 180);
        cout << "Gurvaljin(para) uusev" << endl;
    }

    ~triangle() {
        cout << "Gurvaljin ustav" << endl;
    }

    void print() {
        cout << "Ner: " << get_name() << endl;
        cout << "Koordinatuud: " << endl;
        cout << "a(" << x << ", " << y << ") ";
        cout << "b(" << x1 << ", " << y1 << ") ";
        cout << "c(" << x2 << ", " << y2 << ") " << endl;
        cout << "Urt: " << l << " " << endl;
        cout << "Talbai: " << findArea() << endl;
    }
};

```

```
        cout << "Perimeter: " << findPerimeter() << endl << endl;  
    }  
};
```

Үр дүн:

Durs(para) uusev

2D durs(para) uusev

Toirog(para) uusev

Ner: Toirog

Toirgiin tuv: 10 10

Radius: 5

Talbai: 78.5

Toirgiin urt 31.4

Durs(para) uusev

2D durs(para) uusev

Kvadrat(para) uusev

Ner: Kvadrat

Koordinatuud:

a(10, 10) b(12, 10)

c(12, 12) d(10, 12)

Urt: 2

Talbai: 4

Perimeter: 8

Durs(para) uusev

2D durs(para) uusev

Gurvaljin(para) uusev

Ner: Zuv gurvaljin

Koordinatuud:

a(10, 10) b(13, 16) c(6, 16)

Urt: 8

Talbai: 27.7128

Perimeter: 24

Gurvaljin ustav

2D durs ustav

Durs ustav

Kvadrat ustav

2D durs ustav

Durs ustav

Toirog ustav
2D durs ustav
Durs ustav

5. ДҮГНЭЛТ

Хүүхэд буюу үүсч буй класст байгуулагч тодорхойлон main() функц дотроо объект бий болгон ажлуулж үзэхэд тухайн хүүхэд классын эх класст буй байгуулагч давхар ажиллах бөгөөд энэхүү байгуулагч нь өөрийн гишүүн өгөгдлүүдээ үүсгэж байна. Мөн энэхүү 2 эх класс болон хүүхэд классын байгуулагчдын constructor-ууд дуудагдахад эх классын байгуулагч түрүүлж ажилласан бөгөөд өөрийн гишүүн өгөгдлүүддээ утга оноосны дараачаар хүүхэд классын байгуулагч ажиллаж эхлэв.

6. АШИГЛАСАН МАТЕРИАЛ

➤ “Объект хандлагат программчлал лекцийн материал”, М.Золжаргал

7. ХАВСРАЛТ

Нэгдсэн кодыг хавсаргав.

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

class Shape {
protected:
    char *name;

public:
    void setName(const char *name) {
        if (this->name != nullptr) {
            delete[] this->name;
        }
        this->name = new char[strlen(name) + 1];
        strcpy(this->name, name);
    }

    Shape() {
        this->name = new char[20];
        strcpy(this->name, "default");
        cout << "Durs uusev" << endl;
    }
};
```

```

    }

    Shape(const char *name) {
        this->name = new char[strlen(name) + 1];
        strcpy(this->name, name);
        cout << "Durs(para) uusev" << endl;
    }

    ~Shape() {
        delete[] this->name;
        cout << "Durs ustav" << endl;
    }

    char *get_name() {
        return name;
    }
};

class twoDshape : public Shape {
protected:
    int l;
    int x;
    int y;

public:
    float findArea() {
        return l * l;
    }

    float findPerimeter() {
        return l;
    }

    twoDshape(const char *name) : Shape(name) {
        x = 0;
        y = 0;
        l = 1;
        cout << "2D durs uusev" << endl;
    }
}

```



```

twoDshape(const char *name, int x, int y, int l) : Shape(name) {
    this->x = x;
    this->y = y;
    this->l = l;
    cout << "2D durs(para) uusev" << endl;
}

~twoDshape() {
    cout << "2D durs ustav" << endl;
}

};

class circle : public twoDshape {
public:
    float findArea(){
        return 3.14 * l * l;
    }
    float findPerimeter(){
        return 2 * 3.14 * l;
    }

    circle(const char *name, int x, int y, int l) : twoDshape(name, x, y,
l) {
        cout << "Toirog(para) uusev" << endl;
    }

    ~circle() {
        cout << "Toirog ustav" << endl;
    }

    void print() {
        cout << "Ner: " << get_name() << endl;
        cout << "Toirgiin tuv: " << x << " " << y << endl;
        cout << "Radius: " << l << endl;
        cout << "Talbai: " << findArea() << endl;
        cout << "Toirgiin urt " << findPerimeter() << endl;
    }
};

```

```

class square : public twoDshape {
private:
    int x1, x2, x3, y1, y2, y3;

public:
    float findPerimeter(){
        return 4 * l;
    }

    square(const char *name, int x, int y, int l) : twoDshape(name, x, y,
l) {
        x1 = x + l;
        y1 = y;
        x2 = x + l;
        y2 = y + l;
        x3 = x;
        y3 = y + l;
        cout << "Kvadrat(para) uusev" << endl;
    }

    ~square() {
        cout << "Kvadrat ustav" << endl;
    }

    void print() {
        cout << "Ner: " << get_name() << endl;
        cout << "Koordinatuud: " << endl;
        cout << "a(" << x << ", " << y << ") ";
        cout << "b(" << x1 << ", " << y1 << ") " << endl;
        cout << "c(" << x2 << ", " << y2 << ") ";
        cout << "d(" << x3 << ", " << y3 << ") " << endl;
        cout << "Urt: " << l << " " << endl;
        cout << "Talbai: " << findArea() << endl;
        cout << "Perimeter: " << findPerimeter() << endl << endl;
    }
};

```

```

class triangle : public twoDshape {
private:
    int x1, x2, y1, y2;

```

```

public:
    float findArea(){
        return sqrt(3)/4 *l*l;
    }
    float findPerimeter(){
        return 3 * l;
    }

    triangle(const char *ner, int x, int y, int l) : twoDshape(ner, x, y,
1) {
        y1 = l * cos(30 * 3.14 / 180) + y;
        y2 = l * cos(30 * 3.14 / 180) + y;
        x1 = l * sin(30 * 3.14 / 180) + x;
        x2 = x - l * sin(30 * 3.14 / 180);
        cout << "Gurvaljin(para) uusev" << endl;
    }

    ~triangle() {
        cout << "Gurvaljin ustav" << endl;
    }

    void print() {
        cout << "Ner: " << get_name() << endl;
        cout << "Koordinatuud: " << endl;
        cout << "a(" << x << ", " << y << ") ";
        cout << "b(" << x1 << ", " << y1 << ") ";
        cout << "c(" << x2 << ", " << y2 << ") " << endl;
        cout << "Urt: " << l << " " << endl;
        cout << "Talbai: " << findArea() << endl;
        cout << "Perimeter: " << findPerimeter() << endl << endl;
    }
};

int main() {
    circle circleObj("Toirog", 10, 10, 5);
    circleObj.print();

    square squareObj("Kvadrat", 10, 10, 2);
    squareObj.print();
}

```

```
triangle triangleObj("Zuv gurvaljin", 10, 10, 8);  
triangleObj.print();  
  
return 0;  
}
```