# Класс Удамшуулах Лабораторын Ажил (Лаборатори №7)

# 1. ОРШИЛ/УДИРТГАЛ

Уг лабораторын ажлаар дүрс гэх эх класс тодорхойлж түүнээс удамшсан охин классуудыг тодорхойлон тэдгээр классын объектын талбай пиреметр олох үйлдлүүдийг хийсэн.

# 2. ЗОРИЛГО

Shape гэх эх классыг тодорхойлж түүнээс олон класс удамшуулсан. Түүний тулд дараах зорулгуудыг тавьж ажилласан:

1. Классуудын бүгдэд нь байх ерөнхий шинжийг эх классд тодорхойлсон,

2. Удамшиж буй класс бусад классаас юугаараа ялгаатай байх вэ гэдгийг олж тодорхойлсон,

3. Гишүүн функцуудыг тодорхойлсон.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Класс удамших

Тодорхой нэг классын шинж чанарууд, үйлдлүүдийг өөр нэг класс дотор дахин тодорхойлолгүй шууд авч ашиглах боломжийг удамшил гэж нэрлэдэг. Удамших гэдэг нь байгаа классын шинж үйлдлийг агуулсан шинэ класс үүсэх процесс юм.

## 3.2 Удамшлийн хэлбэр

• Нэг-нэг удамшил (удамших класс нэг эх класстай байна)

• Нэг-олон удамшил (удамших класс 2-с цөөнгүй эх класстай байна)

• Олон түвшинт удамшил (удамших класс удамшсан классаас үүснэ)

• Шаталсан удамшил (нэг классаас олон класс удамших)

• Холимог удамшил (нийлмэл болон шаталсан удамшлийн нийлбэр)

## 3.3 Удамших горим

• Public

• Private

• Protected

Удамшлын горим нь эх класссын гишүүдийг ямар хандалттайгаар авах вэ гэдгийг тодорхойлдог бөгөөд доорх хүснэгтэд үзүүлэв.

|  |  |  |  |
| --- | --- | --- | --- |
| Эх классын  хандалт  Удамших  горим | public | protected | private |
| public | public | protected | Хандахгүй |
| protected | protected | protected | Хандахгүй |
| private | private | protected | Хандахгүй |

Класс удамшуулахдаа

Class Дэд\_класс : удамших\_горим Эх\_класс {

...

};

байдлаар удамшуулна.

# 4. ХЭРЭГЖҮҮЛЭЛТ

Shape классыг тодорхойлсон.

class Shape {

protected:

    char ner[20];

public:

    Shape() {

        strcpy(ner, "durs");

    }

    char\* get\_ner() {

        return ner;

    }

    void set\_ner(char\* a) {

        strcpy(ner, a);

    }

};

Түүнээс удамших TwoDShape классыг тодорхойлж гишүүн өгөгдлийг тодорхойлсон.

class TwoDShape:public Shape {

protected:

    float talbai;

    float pirimetr;

public:

    TwoDShape() {

        strcpy(ner, "durs");

        talbai = 0;

        pirimetr = 0;

    }

    float get\_talbai() {

        return talbai;

    }

    float get\_pirimetr() {

        return pirimetr;

    }

    void set\_talbai(float a) {

        talbai = a;

    }

    void set\_pirimetr(float a) {

        pirimetr = a;

    }

};

TwoDShape классаас Circle, Square, Triangle классуудыг удамшуулж гишүүн өгөгдөл функцуудыг тодорхойлсон.

class Circle:public TwoDShape {

private:

    float Ox, Oy, radius;

public:

    Circle() {

        strcpy(ner, "toirog");

        talbai = 0;

        pirimetr = 0;

        Ox = 0;

        Oy = 0;

        radius = 0;

    }

    float get\_Ox() {

        return Ox;

    }

    float get\_Oy() {

        return Oy;

    }

    float get\_radius() {

        return radius;

    }

    void set\_Ox(float a) {

        Ox = a;

    }

    void set\_Oy(float a) {

        Oy = a;

    }

    void set\_radius(float a) {

        if(a<0) {

            cout << "zov utga oruulna uu!\n";

        } else {

            radius = a;

            talbai = radius\*radius\*M\_PI;

            pirimetr = 2\*M\_PI\*radius;

        }

    }

};

class Square:public TwoDShape {

private:

    float o1x, o1y, o2x, o2y, o3x, o3y, o4x, o4y;

    float taliin\_urt;

public:

    Square() {

        strcpy(ner, "kvadrat");

        talbai = 0;

        pirimetr = 0;

        o1x=0; o2x=0; o3x=0; o4x=0;

        o1y=0; o2y=0; o3y=0; o4y=0;

        taliin\_urt = 0;

    }

    float get\_o1x() {return o1x;}

    float get\_o1y() {return o1y;}

    float get\_o2x() {return o2x;}

    float get\_o2y() {return o2y;}

    float get\_o3x() {return o3x;}

    float get\_o3y() {return o3y;}

    float get\_o4x() {return o4x;}

    float get\_o4y() {return o4y;}

    float get\_taliin\_urt() {return taliin\_urt;}

    void set\_o1x(float a) {

        o1x=a; o2x=a+taliin\_urt; o3x=a; o4x=a+taliin\_urt;

    }

    void set\_o1y(float a) {

        o1y=a; o2y=a; o3y=a-taliin\_urt; o4y=a-taliin\_urt;

    }

    void set\_taliin\_urt(float a) {

        if(a<0) {

            cout << "zov utga oruulna uu!\n";

            talbai = 0;

            pirimetr = 0;

        } else {

            taliin\_urt = a;

            o2x = o1x + taliin\_urt;

            o3y = o1y - taliin\_urt;

            o4x = o1x + taliin\_urt;

            o4y = o1y - taliin\_urt;

            talbai = a\*a;

            pirimetr = 4\*a;

        }

    }

};

class Triangle:public TwoDShape {

private:

    float o1x, o1y, o2x, o2y, o3x, o3y;

    float taliin\_urt;

public:

    float get\_o1x() {return o1x;}

    float get\_o1y() {return o1y;}

    float get\_o2x() {return o2x;}

    float get\_o2y() {return o2y;}

    float get\_o3x() {return o3x;}

    float get\_o3y() {return o3y;}

    float get\_taliin\_urt() {return taliin\_urt;}

    void set\_o1x(float a) {

        o1x=a; o2x=a-taliin\_urt/2; o3x=a+taliin\_urt/2;

    }

    void set\_o1y(float a) {

        o1y=a; o2y=a-taliin\_urt\*sin(M\_PI/3); o3y=a-taliin\_urt\*sin(M\_PI/3);

    }

    void set\_taliin\_urt(float a) {

        if(a<0) {

            cout << "Zov utga oruulna uu!\n";

            talbai = 0;

            pirimetr = 0;

        } else {

            taliin\_urt = a;

            o2x=o1x-taliin\_urt/2;

            o3x=o1x+taliin\_urt/2;

            o2y=o1y-taliin\_urt\*sin(M\_PI/3);

            o3y=o1y-taliin\_urt\*sin(M\_PI/3);

            talbai = taliin\_urt\*taliin\_urt\*sin(M\_PI/3)/2;

            pirimetr = taliin\_urt\*3;

        }

    }

};

Main функц дээр эдгээр классын объектуудыг байгуулж талбай, пиреметрийг олсон.

# 5. ДҮГНЭЛТ

Удамшлыг ашигласнаар хэсэг бүлэг кодыг дахин бичихээс зайлсхийнэ, мөн кодыг дахин ашиглах боломжыг бий болгож байна. Хөгжүүлэгчийн цаг, зардлыг хэмнэнэ. Сул тал нь эх классд алдаатай өөрчлөлт ороход охин классуудад ч мөн нөлөөлнө.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. <https://www.geeksforgeeks.org/inheritance-in-c/?ref=lbp>

# 7. ХАВСРАЛТ

#include <iostream>

#include <string.h>

#include <math.h>

using namespace std;

class Shape {

protected:

    char ner[20];

public:

    Shape() {

        strcpy(ner, "durs");

    }

    char\* get\_ner() {

        return ner;

    }

    void set\_ner(char\* a) {

        strcpy(ner, a);

    }

};

class TwoDShape:public Shape {

protected:

    float talbai;

    float pirimetr;

public:

    TwoDShape() {

        strcpy(ner, "durs");

        talbai = 0;

        pirimetr = 0;

    }

    float get\_talbai() {

        return talbai;

    }

    float get\_pirimetr() {

        return pirimetr;

    }

    void set\_talbai(float a) {

        talbai = a;

    }

    void set\_pirimetr(float a) {

        pirimetr = a;

    }

};

class Circle:public TwoDShape {

private:

    float Ox, Oy, radius;

public:

    Circle() {

        strcpy(ner, "toirog");

        talbai = 0;

        pirimetr = 0;

        Ox = 0;

        Oy = 0;

        radius = 0;

    }

    float get\_Ox() {

        return Ox;

    }

    float get\_Oy() {

        return Oy;

    }

    float get\_radius() {

        return radius;

    }

    void set\_Ox(float a) {

        Ox = a;

    }

    void set\_Oy(float a) {

        Oy = a;

    }

    void set\_radius(float a) {

        if(a<0) {

            cout << "zov utga oruulna uu!\n";

        } else {

            radius = a;

            talbai = radius\*radius\*M\_PI;

            pirimetr = 2\*M\_PI\*radius;

        }

    }

};

class Square:public TwoDShape {

private:

    float o1x, o1y, o2x, o2y, o3x, o3y, o4x, o4y;

    float taliin\_urt;

public:

    Square() {

        strcpy(ner, "kvadrat");

        talbai = 0;

        pirimetr = 0;

        o1x=0; o2x=0; o3x=0; o4x=0;

        o1y=0; o2y=0; o3y=0; o4y=0;

        taliin\_urt = 0;

    }

    float get\_o1x() {return o1x;}

    float get\_o1y() {return o1y;}

    float get\_o2x() {return o2x;}

    float get\_o2y() {return o2y;}

    float get\_o3x() {return o3x;}

    float get\_o3y() {return o3y;}

    float get\_o4x() {return o4x;}

    float get\_o4y() {return o4y;}

    float get\_taliin\_urt() {return taliin\_urt;}

    void set\_o1x(float a) {

        o1x=a; o2x=a+taliin\_urt; o3x=a; o4x=a+taliin\_urt;

    }

    void set\_o1y(float a) {

        o1y=a; o2y=a; o3y=a-taliin\_urt; o4y=a-taliin\_urt;

    }

    void set\_taliin\_urt(float a) {

        if(a<0) {

            cout << "zov utga oruulna uu!\n";

            talbai = 0;

            pirimetr = 0;

        } else {

            taliin\_urt = a;

            o2x = o1x + taliin\_urt;

            o3y = o1y - taliin\_urt;

            o4x = o1x + taliin\_urt;

            o4y = o1y - taliin\_urt;

            talbai = a\*a;

            pirimetr = 4\*a;

        }

    }

};

class Triangle:public TwoDShape {

private:

    float o1x, o1y, o2x, o2y, o3x, o3y;

    float taliin\_urt;

public:

    float get\_o1x() {return o1x;}

    float get\_o1y() {return o1y;}

    float get\_o2x() {return o2x;}

    float get\_o2y() {return o2y;}

    float get\_o3x() {return o3x;}

    float get\_o3y() {return o3y;}

    float get\_taliin\_urt() {return taliin\_urt;}

    void set\_o1x(float a) {

        o1x=a; o2x=a-taliin\_urt/2; o3x=a+taliin\_urt/2;

    }

    void set\_o1y(float a) {

        o1y=a; o2y=a-taliin\_urt\*sin(M\_PI/3); o3y=a-taliin\_urt\*sin(M\_PI/3);

    }

    void set\_taliin\_urt(float a) {

        if(a<0) {

            cout << "Zov utga oruulna uu!\n";

            talbai = 0;

            pirimetr = 0;

        } else {

            taliin\_urt = a;

            o2x=o1x-taliin\_urt/2;

            o3x=o1x+taliin\_urt/2;

            o2y=o1y-taliin\_urt\*sin(M\_PI/3);

            o3y=o1y-taliin\_urt\*sin(M\_PI/3);

            talbai = taliin\_urt\*taliin\_urt\*sin(M\_PI/3)/2;

            pirimetr = taliin\_urt\*3;

        }

    }

};

main() {

    Circle a;

    Square b;

    Triangle c;

    float n,m;

    char ner[20];

    cout << "Toirgiin neriig oruul: ";

    cin >> ner;

    a.set\_ner(ner);

    cout << "Toirgiin toviin koordinatiig oruul: ";

    cin >> n >> m;

    a.set\_Ox(n);

    a.set\_Oy(m);

    cout << "Toirgiin radiusiig oruul: ";

    cin >> n;

    a.set\_radius(n);

    cout << "ner: " << a.get\_ner() << endl;

    cout << "tov tsegiin koordinat: " << a.get\_Ox() << " " << a.get\_Oy() << endl;

    cout << "radius: " << a.get\_radius() << endl;

    cout << "talbai: " << a.get\_talbai() << endl;

    cout << "pirimetr: " << a.get\_pirimetr() << endl;

    cout << "Kvadratiin neriig oruul: ";

    cin >> ner;

    b.set\_ner(ner);

    cout << "Kvadratiin zuun deed oroin koordinatiig oruul: ";

    cin >> n >> m;

    b.set\_o1x(n);

    b.set\_o1y(m);

    cout << "Kvadratiin taliin urtiig oruul: ";

    cin >> n;

    b.set\_taliin\_urt(n);

    cout << "ner: " << b.get\_ner() << endl;

    cout << "zuun deed oroin koordinat: " << b.get\_o1x() << " " << b.get\_o1y() << endl;

    cout << "zuun dood oroin koordinat: " << b.get\_o3x() << " " << b.get\_o3y() << endl;

    cout << "baruun deed oroin koordinat: " << b.get\_o2x() << " " << b.get\_o2y() << endl;

    cout << "baruun dood oroin koordinat: " << b.get\_o4x() << " " << b.get\_o4y() << endl;

    cout << "taliin urt: " << b.get\_taliin\_urt() << endl;

    cout << "talbai: " << b.get\_talbai() << endl;

    cout << "pirimetr: " << b.get\_pirimetr() << endl;

    cout << "Gurvaljnii neriig oruul: ";

    cin >> ner;

    c.set\_ner(ner);

    cout << "Gurvaljnii deed oroin koordinatiig oruul: ";

    cin >> n >> m;

    c.set\_o1x(n);

    c.set\_o1y(m);

    cout << "Gurvaljnii taliin urtiig oruul: ";

    cin >> n;

    c.set\_taliin\_urt(n);

    cout << "ner: " << c.get\_ner() << endl;

    cout << "deed oroin koordinat: " << c.get\_o1x() << " " << c.get\_o1y() << endl;

    cout << "zuun dood oroin koordinat: " << c.get\_o2x() << " " << c.get\_o2y() << endl;

    cout << "baruun dood oroin koordinat: " << c.get\_o3x() << " " << c.get\_o3y() << endl;

    cout << "taliin urt: " << c.get\_taliin\_urt() << endl;

    cout << "talbai: " << c.get\_talbai() << endl;

    cout << "pirimetr: " << c.get\_pirimetr() << endl;

}