

Лаборатори 9:

1. Хийсвэр функц гэж юу вэ?

Хийсвэр функц нь эх класс дотор зарлагдсан, үүссэн классаар дахин тодорхойлогдсон (overriden) гишүүн функцийг хэлнэ. Хийсвэр функцийг зөвхөн эх класс дотор зарлана. Эх ба удамших классууд ижил нэртэй функцтэй байвал эх классынхыг нь virtual гэж зарласнаар эх ба удамших классын аль нь ч функцээ дуудах боломжтой болдог.

Эх классыг зарлахдаа virtual түлхүүр үгийг ашиглан

```
virtual void functionName(){
```

```
...
```

```
};
```

- Хийсвэр функц нь эх классын кодыг сольж програмчлахыг удамших класст зөвшөөрдөг.
- Эх классын байгуулагч функцээс бусад нь хийсвэр байж болно.

2. Жинхэнэ хийсвэр функц гэж юу вэ?

Функцын тодорхойлолтгүй зөвхөн зарласан функцийг жинхэнэ хийсвэр функц гэнэ. Жинхэнэ хийсвэр функцийг эх классаас удамшсан хүүхэд класст тодорхойлж өгдөг.

Тодорхойлолт:

(Эх класс) virtual float area() = 0;

```
(Хүүхэд класс) float area() {  
    float area = 3.14 * rad * rad;  
    return area;  
}
```

3. Функц дахин программчлах гэж юу вэ? Эх классын дахин программчилсан функцийг хүүхэд классын функц дотроос хэрхэн дууддаг вэ?

Эх класс дахь гишүүн функцийг нэр нь яг адил байхаар дахин тодорхойлохыг функц дахин програмчлах гэнэ. Дахин програмчлах удамших классын арга нь бичлэгдсэн хувьд эх класстай адилхан байх ёстой.

Эх классын дахин программчилсан функцийг хүүхэд классын функц дотроос хүүхэд классаас үүсгэсэн объектоор нь дамжуулж хандана.

4. Хийсвэр класс гэж юу вэ? Хэрхэн объект байгуулдаг вэ?

Жинхэнэ хийсвэр функц агуулсан классыг abstract буюу хийсвэр класс гэнэ. Ийм классын объектыг үүсгэж болохгүй. Харин түүнээс удамшсан класст жинхэнэ хийсвэр функцийг заавал дахин програмчлахыг шаарддаг

ба хэрэв дахин програмчласан тохиолдолд хүүхэд классын объектыг үүсгэж болно.

5. Удамшилд байгуулагч функц хэрхэн ашиглагддаг вэ? Удамшил ба байгуулагч функцийг хүрээнд үзсэн зүйлүүдийг бич.

Удамшсан классын объектыг байгуулахад эхлээд эх классын анхдагч байгуулагч дуудагдсаны дараа хүүхэд классын байгуулагч дуудагддаг. Ямар ч тохиолдолд эх классын анхдагч байгуулагч дуудагддаг.

6. Удамшил ба устгагч функц хоёр ямар хамааралтай вэ?

Хүүхэд классын объект устахад эхлээд хүүхэд классын устгагч, дараагаар нь эх классын устгагч тус тус дуудагдан ажилладаг. Дарааллын хувьд удамшил нь эхлээд дараагаар нь эх класс нь дуудагдана.

```
#define pi 3.14
class Shape
{
protected:
    char *name;

public:
    virtual float perimeter() = 0;

public:
    Shape()
    {
        name = new char[2];
        strcpy(name, "");
    }
    Shape(char *n)
    {
        name = new char[strlen(n) + 1];
        strcpy(name, n);
    }

    ~Shape()
    {
        delete name;
    }
};

class TwoDiShape : public Shape
{
protected:
```

```

    int *x;
    int *y;

public:
    virtual float area() = 0;
    virtual void inputData() = 0;
};

class Circle : public TwoDiShape
{
private:
    float rad;

public:
    Circle()
    {
        name = new char[6];
        strcpy(name, "Circle");
        x = new int[1];
        y = new int[1];
        rad = 0;
    }
    ~Circle()
    {
        delete name;
        delete x;
        delete y;
    }

    void setX(int _x)
    {
        x = &_x;
    }
    void setY(int _y)
    {
        y = &_y;
    }
    void setRad(int r)
    {
        rad = r;
    }

    float area()
    {
        return pi * rad * rad;
    }
    float perimeter()
    {

```

```

        return 2 * pi * rad;
    }
void inputData()
{
    int x, y;
    cout << "Radius: ";
    cin >> rad;
    cout << "x1: ";
    cin >> x;
    setX(x);
    cout << "y1: ";
    cin >> y;
    setX(y);
}
};

class Square : public TwoDiShape
{
private:
    float a;

public:
    Square()
    {
        name = new char[6];
        strcpy(name, "Square");
        x = new int[4];
        y = new int[4];
        a = 0;
    }

    ~Square()
    {
        delete name;
        delete x;
        delete y;
    }
    void setX(int _x)
    {
        x[0] = _x;
        x[1] = _x + a;
        x[2] = _x + a;
        x[3] = _x;
    }
    void setY(int _y)
    {
        y[0] = _y;
        y[1] = _y;

```

```

        y[2] = _y - a;
        y[3] = _y - a;
    }

    void setA(int side)
    {
        a = side;
    }

    float area()
    {
        return a * a;
    }
    float perimeter()
    {
        return 4 * a;
    }
    void inputData()
    {
        int x, y;
        cout << "Taliin urt: ";
        cin >> a;
        cout << "x1: ";
        cin >> x;
        setX(x);
        cout << "y1: ";
        cin >> y;
        setX(y);
    }
};

class Triangle : public TwoDiShape
{
private:
    float a;

public:
    Triangle()
    {
        name = new char[8];
        strcpy(name, "Triangle");
        x = new int[3];
        y = new int[3];
        a = 0;
    }

    ~Triangle()
    {

```

```

        delete name;
        delete x;
        delete y;
    }
    void setX(int _x)
    {
        x[0] = _x;
        x[1] = _x + a / 2;
        x[2] = _x - a / 2;
    }
    void setY(int _y)
    {
        float undur = sqrt(3) / 4 * a;
        y[0] = _y;
        y[1] = _y - undur;
        y[2] = _y - undur;
    }

    void setA(int side)
    {
        a = side;
    }
    float area()
    {
        return sqrt(3) / 4 * a * a;
    }
    float perimeter()
    {
        return 3 * a;
    }
    void inputData()
    {
        int x, y;
        cout << "Taliin urt: ";
        cin >> a;
        cout << "x1: ";
        cin >> x;
        setX(x);
        cout << "y1: ";
        cin >> y;
        setX(y);
    }
};

int main()
{
    int n = 7;
    TwoDiShape *shape[n];

```

```

shape[0] = new Circle;
shape[0]→inputData();
shape[1] = new Circle;
shape[1]→inputData();
shape[2] = new Square;
shape[2]→inputData();
shape[3] = new Square;
shape[3]→inputData();
shape[4] = new Square;
shape[4]→inputData();
shape[5] = new Triangle;
shape[5]→inputData();
shape[6] = new Triangle;
shape[6]→inputData();

TwoDiShape *temp;
for (int i = 0; i < n - 1; i++)
{
    for (int j = 0; j < n - i - 1; j++)
    {
        if (shape[j]→area() < shape[j + 1]→area())
        {
            temp = shape[j];
            shape[j] = shape[j + 1];
            shape[j + 1] = temp;
        }
    }
}

cout << "Dursuudiin eremblegdsen talbai: \n";
for (int i = 0; i < n; i++)
{
    cout << shape[i]→area() << endl;
}

return 0;
}

```