

ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ  
МЭДЭЭЛЭЛ, ХОЛБООНЫ ТЕХНОЛОГИЙН СУРГУУЛЬ

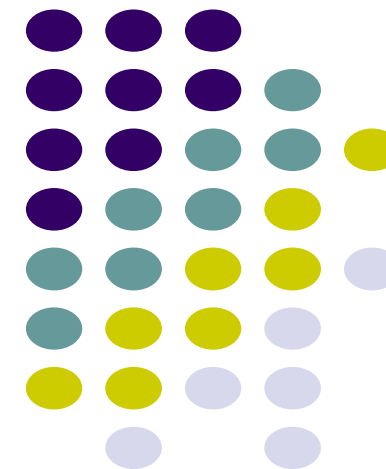
F.CS202  
ОБЪЕКТ ХАНДЛАГАТ ПРОГРАМЧЛАЛ

Лекц №13

**Полиморфизм – классын олон  
хэлбэржилт**

док., дэд проф. Б.Батзолбоо  
маг. Б.Мөнхбуян

2021 он



# Агуулга



- Полиморфизм гэж юу вэ?
- Удамшлын харьцааны тухай
- Массив ашиглан полиморфизм хэрэгжүүлэх

# Полиморфизм



# Полиморфизм



Ижил үйлдлээр ялгаатай үр дүнд хүрэх

- Compile time: Метод дахин тодорхойлох
- Runtime: Метод дарж тодорхойлох

# Compile time Полиморфизм Жишээ



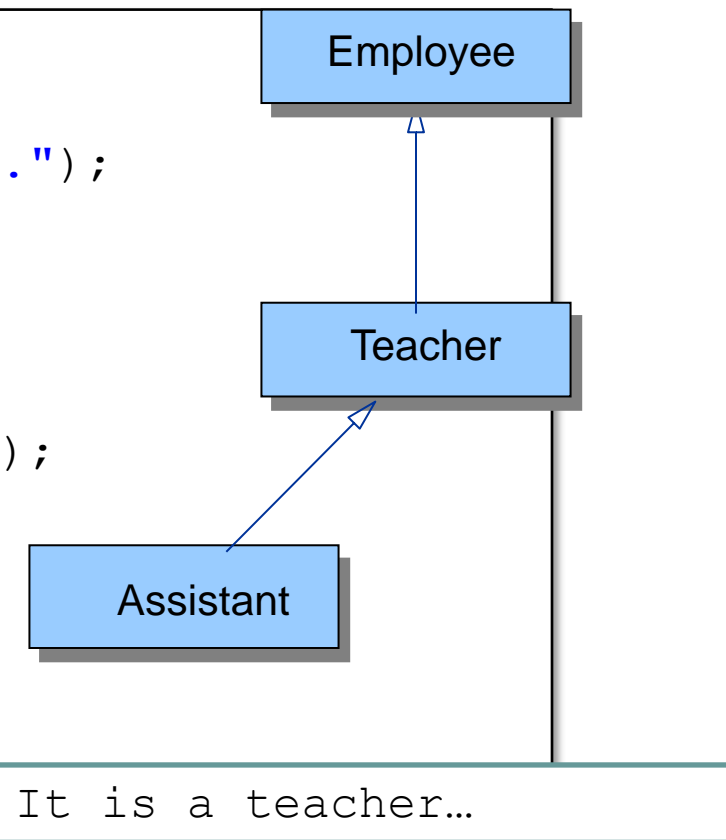
```
class MultiplyFun {
    static int Multiply(int a, int b)
    {
        return a * b;
    }
    static double Multiply(double a, double b)
    {
        return a * b;
    }
}
class MainProg {
    public static void main(String[] args)
    {
        System.out.println(MultiplyFun.Multiply(4, 12));

        System.out.println(MultiplyFun.Multiply(12.12, 2.1));
    }
}
```

# Runtime Полиморфизм Жишээ



```
class Employee{
    void info(){
        System.out.println("It is an employee...");
    }
}
class Teacher extends Employee{
    void info(){
        System.out.println("It is a teacher...");
    }
}
class Assistant extends Teacher{
    public static void main(String args[]){
        Employee e=new Assistant();
        e.info();
    }
}
```

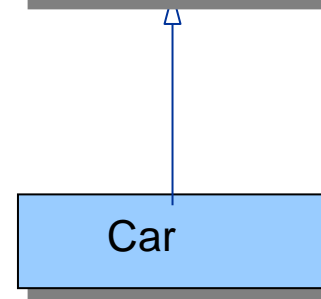
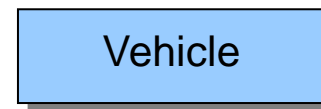


# Runtime Полиморфизм Жишээ



```
class Vehicle{
    int seats=2;
}

class Car extends Vehicle{
    int seats=4;
    public static void main(String args[]){
        Vehicle v=new Car();
        System.out.println(v.seats);
    }
}
```



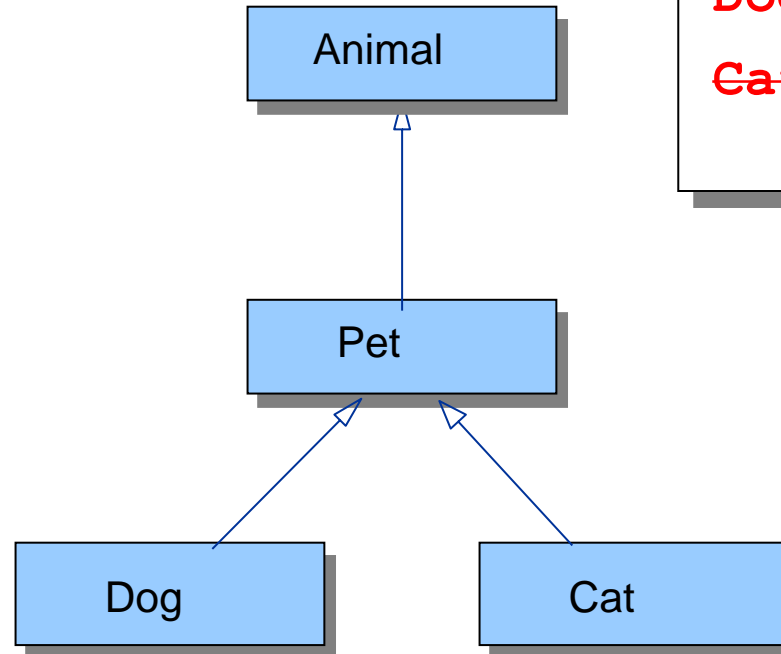
2



# Полиморфизм

- **Полиморфизм** нь нэг хувьсагчаар удамшлын шатлал дээрх өөр дэд классуудаас үүссэн объектуудыг заах боломжийг олгодог.

```
Pet myPet;  
myPet = new Dog();  
myPet = new Cat();  
  
Animal myAnimal;  
myAnimal = new Dog();  
myAnimal = new Cat();  
  
myAnimal = myPet;
```



```
Dog myDog = new Cat();  
Cat myCat = new Pet();
```

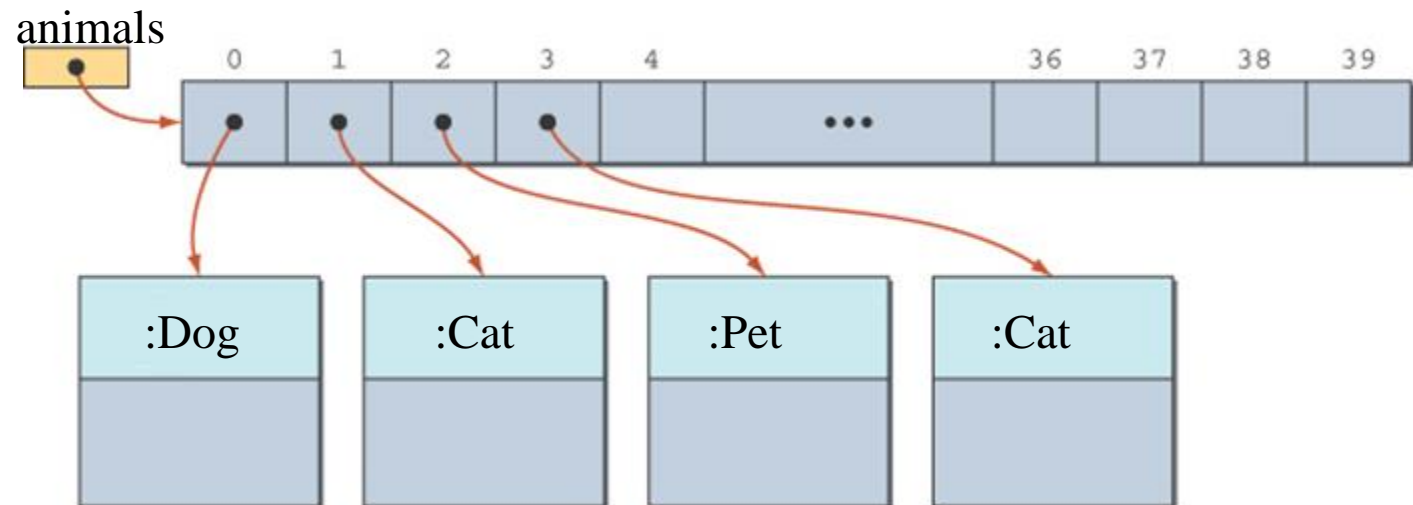




# Массив үүсгэх

```
Animal animals = new Animals[40];  
.  
.  
.  
animals[0] = new Dog();  
animals[1] = new Cat();  
animals[2] = new Pet();  
animals[3] = new Cat();  
.  
.  
.
```

animals массивын элементүүд  
Dog, Cat, Pet классуудын  
объектуудын заагч байна.



# Классууд



```
public class Pet {  
    private String name;  
    public String getName() { return name; }  
    public void setName(String newName) { name=newName; }  
    public String speak() {  
        return name + ":I'm pet";  
    }  
}
```

```
}  
public class Dog extends Pet {  
    public String speak() {  
        return name + ":How how";  
    }  
}
```

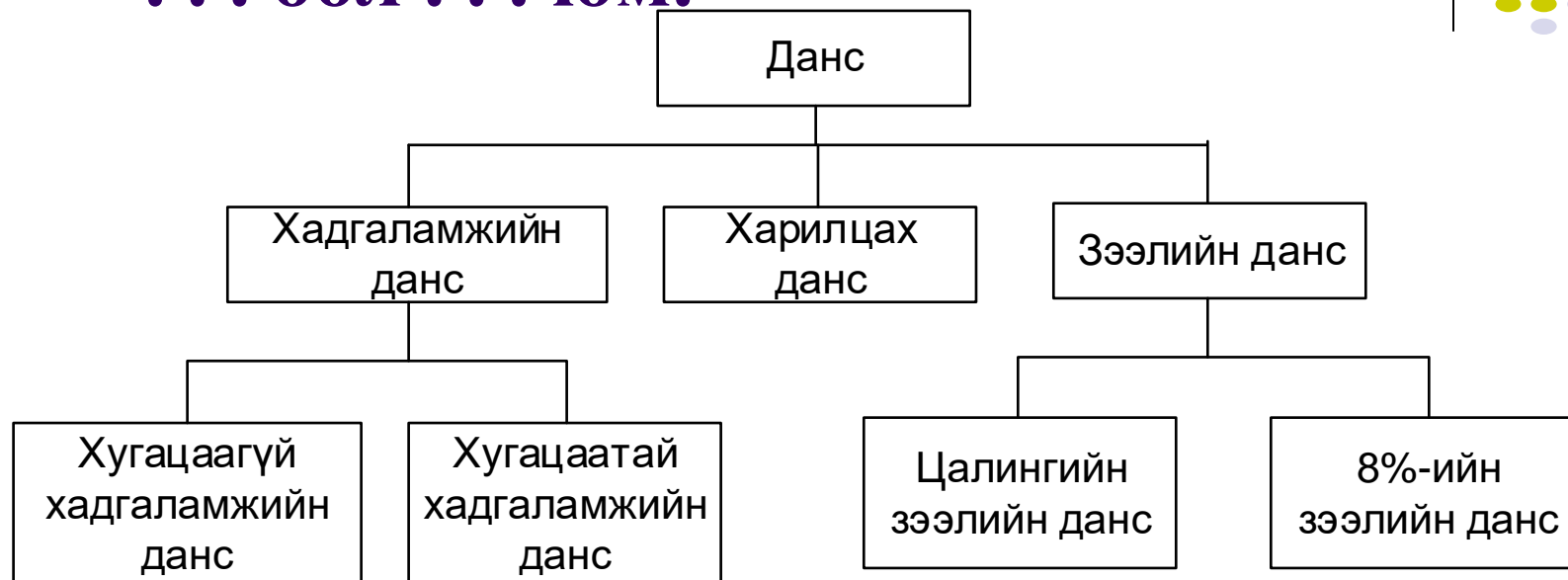
```
public class Cat extends Pet {  
    public String speak() {  
        return name + ":Mia mia";  
    }  
}
```

# Полиморфик мэссэж илгээх



- Бүх амьтдыг дуудах

```
for (int i = 0; i < animals.length; i++) {  
    animals[i].speak();  
}
```



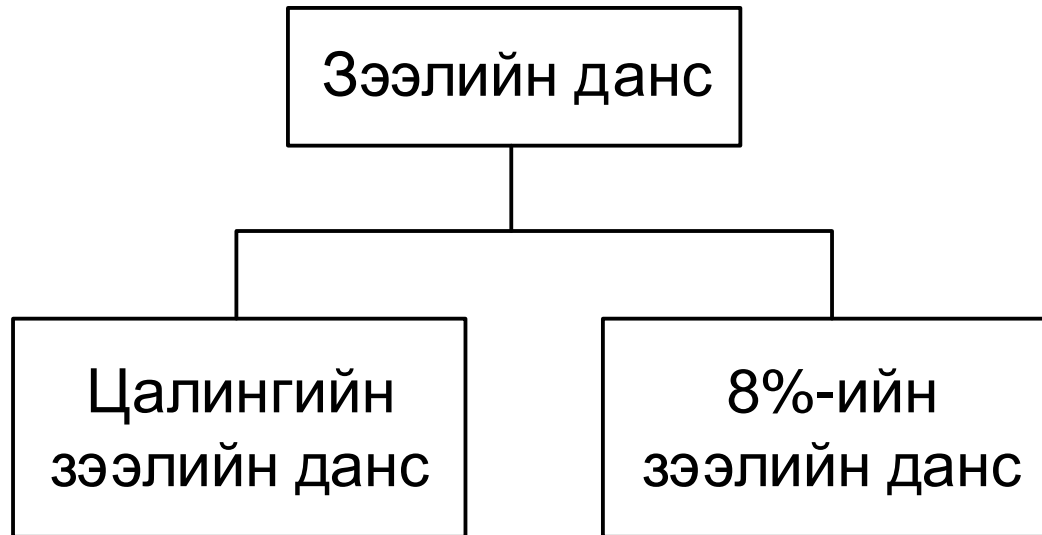
а451234567 бол хугацаагүй хадгаламжийн данс классын тохиолдол байхад:

- а451234567 бол Хугацаагүй хадгаламжийн данс юм.
- Хугацаагүй хадгаламжийн данс бол Хадгаламжийн данс юм.
- а451234567 бол Хадгаламжийн данс юм.
- Хадгаламжийн данс бол Данс юм.
- Хугацаагүй хадгаламжийн данс бол Данс юм.
- а451234567 бол Данс юм.



# Объект үүсгэх жишээ

```
ЗээлийнДанс зээлийнДанс[] = new ЗээлийнДанс[2];  
зээлийнДанс[0] = new ЦалингийнЗээлийнДанс();  
зээлийнДанс[1] = new НайманЗээлийнДанс();
```



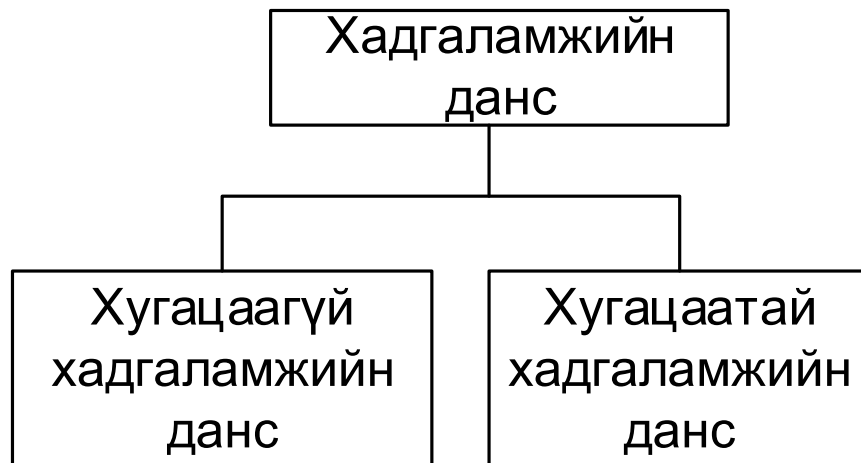
# Полиморфизмын жишээ



```
ХадгаламжийнДанс[] хадДанс = new ХадгаламжийнДанс[2];
```

```
хадДанс[0] = new ХугацаагүйХадгамжийнДанс();
```

```
хадДанс[1] = new ХугацаатайХадгамжийнДанс();
```

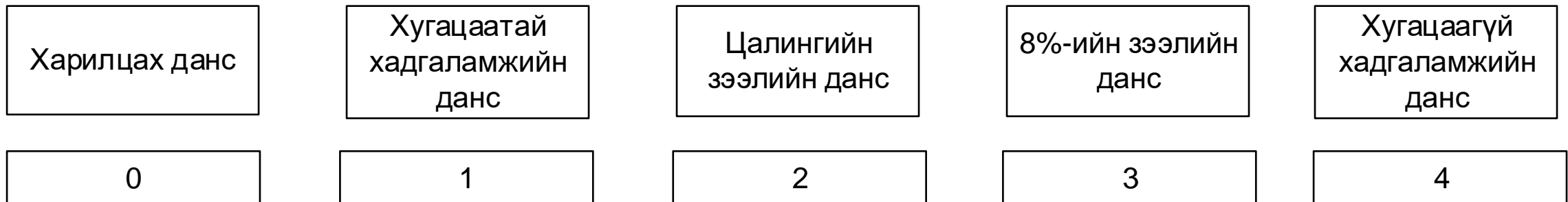




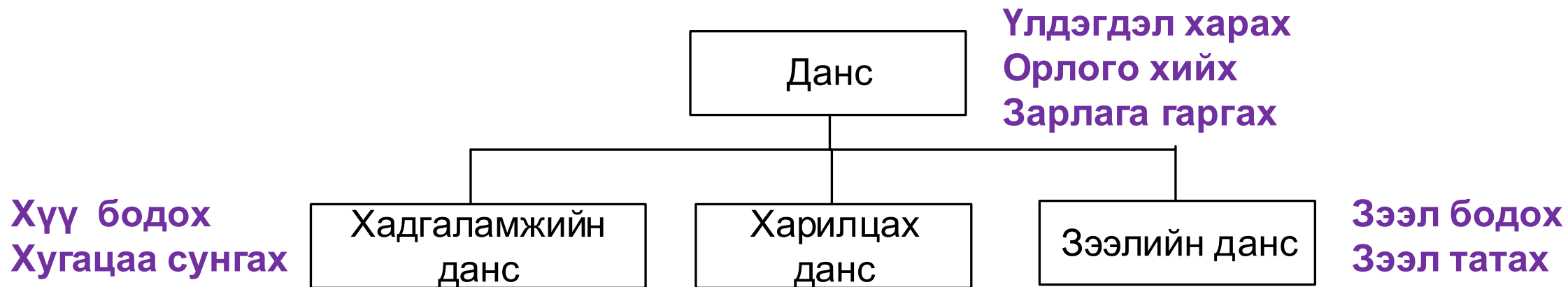
# Полиморфизмын жишээ

```
Данс минийДанснууд[] = new Данс[5]
минийДанснууд[0] = new ХарилцахДанс();
минийДанснууд[1] = new ХугацаатайХадгаламжийнДанс();
минийДанснууд[2] = new ЦалингийнЗээлийнДанс();
минийДанснууд[3] = new НайманЗээлийнДанс();
минийДанснууд[4] = new ХугацаагүйХадгаламжийнДанс();
```

```
Данс МинийДанс = new ЦалингийнЗээлийнДанс();
```



# Классын үйлдлүүд







# Массивын элементүүдийн үйлдлийг дуудах

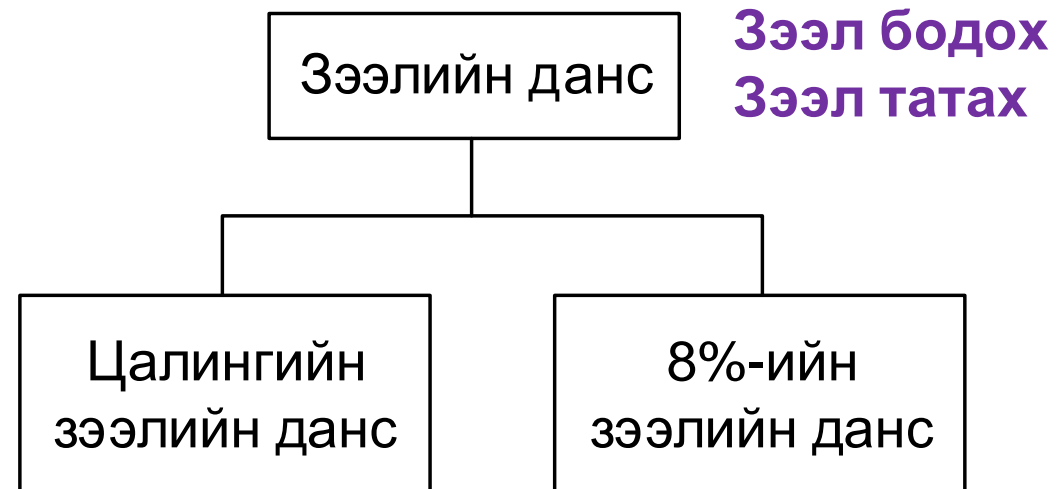
```
ЗээлийнДанс[] зээлийнДанс = new ЗээлийнДанс[2];
```

```
зээлийнДанс[0] = new ЦалингийнЗээлийнДанс();
```

```
зээлийнДанс[1] = new НайманЗээлийнДанс();
```

```
for (int i=0;i<2;i++)
```

```
    зээлийнДанс[i].зээлБодох();
```





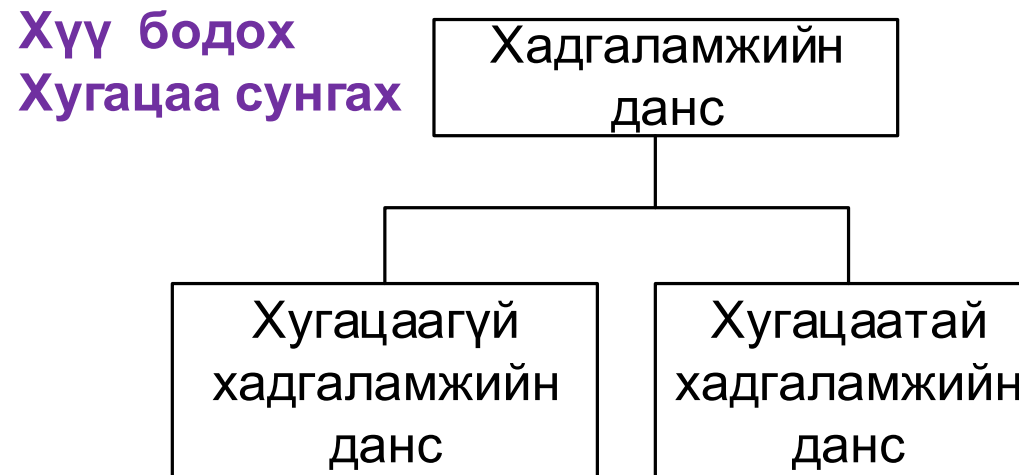
# Полиморфизмын давуу

```
ХадгаламжийнДанс хадДанс[] = new ХадгаламжийнДанс[2];
```

```
хадДанс[0] = new ХугацаагүйХадгамжийнДанс();
```

```
хадДанс[1] = new ХугацаатайХадгамжийнДанс();
```

```
for (int i=0;i<2;i++)  
    хадДанс[i].хүүБодох();
```





# Полиморфизмын давуу

```
Данс минийДанснууд[] = new Данс[5]
минийДанснууд[0] = new ХарилцахДанс();
минийДанснууд[1] = new ХугацаатайХадгаламжийнДанс();
минийДанснууд[2] = new ЦалингийнЗээлийнДанс();
минийДанснууд[3] = new НайманЗээлийнДанс();
минийДанснууд[4] = new ХугацаагүйХадгаламжийнДанс();
```

```
нийтҮлдэгдэлийнХэмжээ() {
```

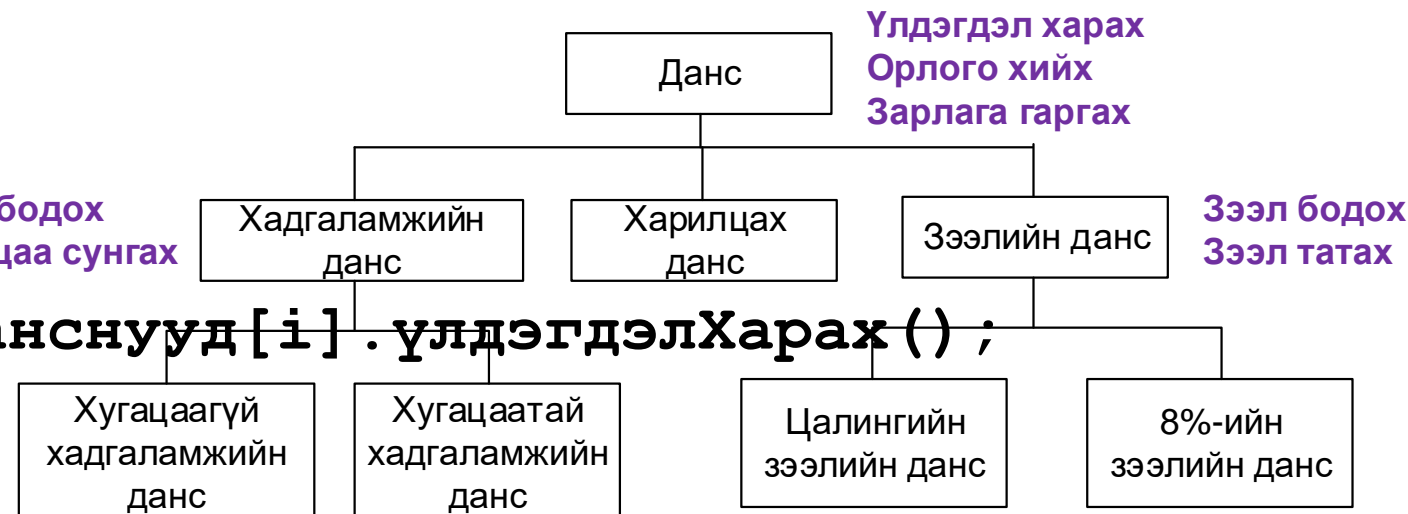
```
    нийтҮлдэгдэл = 0;
```

```
    for (int i=0; i<5; i++)
```

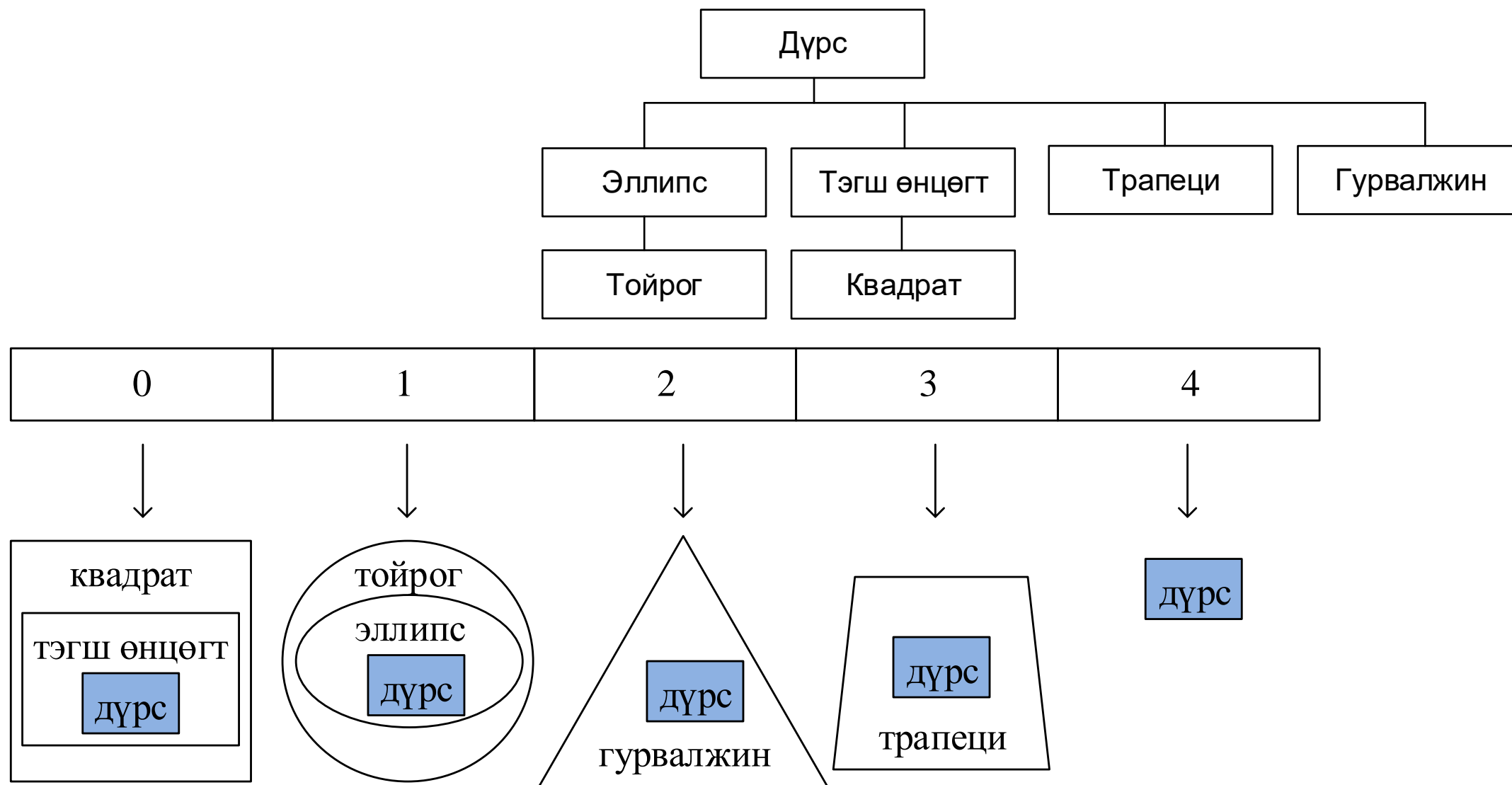
```
        нийтҮлдэгдэл += минийДанснууд[i].үлдэгдэлХарах();
```

```
    return нийтҮлдэгдэл;
```

```
}
```



# Полиморфизмын өөр жишээ

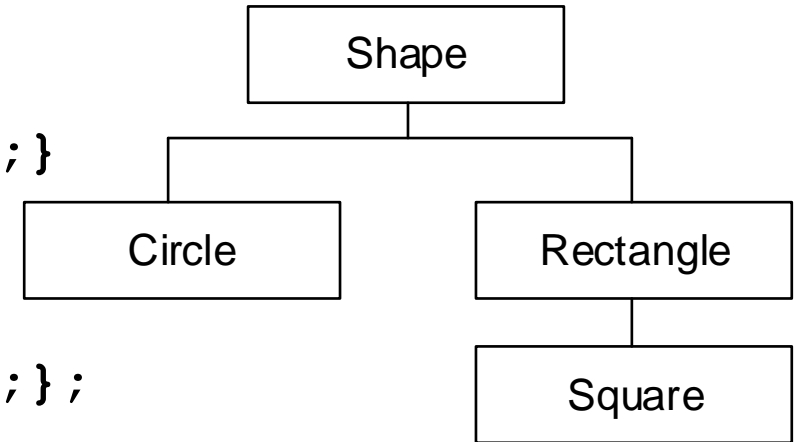


# Жишээ код



```
public abstract class Shape {  
    protected String shapeName;  
    public Shape(String name) {shapeName = name;}  
    public abstract double area ( );  
    public abstract double perimeter ( );  
    public String toString( ) {return shapeName;};  
}
```

```
public class Rectangle extends Shape {  
    protected double length, width;  
    public Rectangle(double len, double wid) {  
        super("Rectangle");  
        length = len;    width = wid;    }  
    public double area ( ) {return length * width;}  
    public double perimeter ( ) {  
        return 2.0 * (length + width);    }  
}
```



## Жишээ код



```
public class Circle extends Shape {  
    private double radius;  
    public Circle (double rad) {  
        super ("Circle");  
        radius = rad;  
    }  
    public double area( ) {  
        return Math.PI * radius * radius; }  
    public double perimeter( ) {  
        return 2.0 * Math.PI * radius; }  
}  
  
public class Square extends Rectangle {  
    public Square (double side) {  
        super ("Square");  
        length = width = side;  
    }  
}
```

Rectangle классын  
гишүүд

# Дүрсүүдийн массив



```
public class ShapeShifter {  
    public static void main (String [ ] args) {  
        Shape [ ] shapeList = new Shape[5];  
        shapeList[0] = new Circle(3.0);  
        shapeList[1] = new Rectangle(3.0, 4.0);  
        shapeList[2] = new Rectangle(2.5, 7.5);  
        shapeList[3] = new Circle(2.5);  
        shapeList[4] = new Square(5.0);  
        for (int i = 0; i < shapeList.length; i++) {  
            System.out.print (shapeList[i].toString( ) + "    " );  
            System.out.print (shapeList[i].area( ) + "    ");  
            System.out.println (shapeList[i].perimeter( ));  
        }  
    }  
}
```

Shape (болон түүнээс  
удамшсан) классын  
объектуудыг хадгалах массив

Классуудаас үүссэн  
объектуудыг массивт хадгалах

Массивын элементүүд дээр  
давтаж, талбай болон  
периметр бодож харуулах

# ДҮГНЭЛТ



- Полиморфизм нь объект хандлагат програмчлалын тулгуур ойлголтын нэг юм
- Полиморфизм нь ижил үйлдлээр ялгаатай үр дүнд хүрэх
- Полиморфизмын жишээг удамшлын модон дээр тайлбарлалаа
- Массив ашиглан полиморфизм хэрэгжүүлэх.