

Объектно-ориентированное программирование

Алексей Владыкин

СПбГУ ИТМО

09 ноября 2009

Основные понятия

- Класс — общее описание совокупности объектов
- Объект — экземпляр класса

- Инкапсуляция — сокрытие деталей реализации
- Наследование — возможность создания подклассов
- Полиморфизм — исполнение разного кода в зависимости от класса объекта

Пример: класс Rectangle

```
public class Rectangle {
    private double x;
    private double y;
    private double w;
    private double h;
    public Rectangle(double _x, double _y, double _w, double _h) {
        x = _x;
        y = _y;
        w = _w;
        h = _h;
    }
    public double getX() {
        return x;
    }
    public double getY() {
        return y;
    }
    public double getArea() {
        return w * h;
    }
}
```

Интерфейс, абстрактный класс

- Класс определяет «внешний вид» объекта и его внутреннее устройство
- Интерфейс определяет внешний вид объекта, но не его устройство
- Абстрактный класс — гибрид объекта и интерфейса; определяет внешний вид и частично устройство

Пример: интерфейс Shape

```
public interface Shape {  
    public double getX();  
    public double getY();  
    public double getArea();  
}
```

```
public class Rectangle implements Shape {  
    // ...  
}
```

Пример: класс AbstractShape

```
public abstract class AbstractShape implements Shape {
    private double x;
    private double y;
    public AbstractShape(double _x, double _y) {
        x = _x;
        y = _y;
    }
    public double getX() {
        return x;
    }
    public double getY() {
        return y;
    }
    public abstract double getArea();
}
```

Пример: классы Rectangle и Circle

```
public class Rectangle extends AbstractShape {
    private double w;
    private double h;
    public Rectangle(double _x, double _y, double _w, double _h) {
        super(_x, _y);
        w = _w;
        h = _h;
    }
    public double getArea() { return w * h; }
}
```

```
public class Circle extends AbstractShape {
    private double r;
    public Circle(double _x, double _y, double _r) {
        super(_x, _y);
        r = _r;
    }
    public double getArea() { return Math.PI * r * r; }
}
```

Пример: полиморфизм в действии

```
public class ShapeTest {
    public static void main(String [] args) {
        Shape r = new Rectangle(0, 0, 10, 20);
        System.out.printf(
            "Shape: x=%f, y=%f, area=%f\n",
            r.getX(), r.getY(), r.getArea());

        Shape c = new Circle(5, 5, 4);
        System.out.printf(
            "Shape: x=%f, y=%f, area=%f\n",
            c.getX(), c.getY(), c.getArea());
    }
}
```