



Интерфейсы



Определение интерфейсов



Интерфейсы в Java - это способ определить набор методов, которые класс должен реализовать.

Интерфейс похож на контракт: если класс подписывает этот контракт (то есть реализует интерфейс), он обязуется предоставить конкретные реализации для всех методов интерфейса.

Интерфейсы не содержат реализации методов (за исключением default-методов, о которых пойдет речь ниже); они только определяют "форму" или "шаблон" методов.



Контрольная точка

- Понятно ли что такое интерфейсы?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



Иерархии наследования с абстрактными классами

```
package l27.ex1;

public interface Drawable {
    // Метод, который должен быть реализован
    void draw();
}

class Circle implements Drawable {
    @Override
    public void draw() {
        System.out.println("Рисуем круг");
    }
}

class Square implements Drawable {
    @Override
    public void draw() {
        System.out.println("Рисуем квадрат");
    }
}
```

Drawable - интерфейс.

Circle и Squeare - классы которые реализуют интерфейс Drawable.



Контрольная точка

- Понятно ли как писать и использовать интерфейсы в самом простом варианте?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



Множественная имплементация интерфейсов

```
package 127.ex2;

import 127.ex1.Drawable;

public interface Fillable {
    void fill();
}

// Класс Circle теперь реализует два интерфейса:
// Drawable и Fillable
class Circle implements Drawable, Fillable {
    @Override
    public void draw() {
        System.out.println("Рисуем круг");
    }

    @Override
    public void fill() {
        System.out.println("Закрашиваем круг");
    }
}
```

Fillable - новый интерфейс.

Circle - класс, который **реализуют** интерфейсы Drawable и Fillable.



Контрольная точка

- Понятно ли что такое множественная имплементация интерфейсов и как ее использовать?

Если все ясно, ставим плюсы, иначе - задаем вопросы.



Default-методы в интерфейсах

```
package l27.ex3;

interface Drawable {
    void draw();

    default void resize() {
        System.out.println("Изменяем размер");
    }
}

class Circle implements Drawable {
    @Override
    public void draw() {
        System.out.println("Рисуем круг");
    }

    // Метод resize не обязательно
    // переопределять
}
```

С Java 8 в интерфейсах появилась возможность определять default-методы с реализацией.

Это позволяет добавлять новые функции в интерфейсы без нарушения существующего кода, который эти интерфейсы реализует.

Default-методы можно переопределять в классах, реализующих интерфейс, но это необязательно.



Домашнее задание

Домашнее задание:

Задача минимум: прочитать статью <https://metanit.com/java/tutorial/3.7.php>

Задача минимум:

Придумать контракт (взаимодействия) для своих сущностей и реализовать на их основе соответствующие интерфейсы

- Вариант 1: на основе 24 домашки основанной на 17 домашке
- Вариант 2: на основе 24 домашки основанной на Покемонах
- Вариант 3: на основе 26 домашки основанной на фигурах

Задача максимум: сделать все вышесказанное, но использовать все особенности интерфейсов упомянутые в презентации и статье.



The end