Java Fundamentals

Классы и объекты

WebCamp

Классы и объекты

Объектно-ориентированный подход

Парадигмы ООП

Инкапсуляция — механизм языка, позволяющий ограничить доступ одних компонентов программы к другим.

Наследование —механизм позволяющий описать новый класс на основе уже существующего (родительского), при этом свойства и функциональность родительского класса заимствуются новым классом.

Полиморфизм —возможность объектов с одинаковой спецификацией иметь различную реализацию.

Абстракция в ООП – это придание объекту характеристик, которые отличают его от всех других объектов, четко определяя его концептуальные границы.

Абстрагирование в ООП – это способ выделить набор значимых характеристик объекта, исключая из рассмотрения незначимые. Соответственно, абстракция —это набор всех таких характеристик.

Ключевые слова

Ключевое слово this

this – ключевое слово, которое является ссылкой на ТЕКУЩИЙ ОБЪЕКТ.

```
public Car(String model, int maxSpeed, int speed) {
      this.model = model;
      this.maxSpeed = maxSpeed;
      this.speed = speed;
  }
```

В конструкторе инициализируются поля текущего объекта значениями, переданными при вызове конструктора

Конструкторы

Конструктор – специальный метод для создания экземпляра класса (инициализации объекта).

Имя конструктора соответствует имени класса.

```
public Car(String model, int maxSpeed, int speed) {
      this.model = model;
      this.maxSpeed = maxSpeed;
      this.speed = speed;
  }
```

Конструкторы

Конструктор по умолчанию

Если программист явно не задал конструктор в классе, JVM при компиляции создаст конструктор по умолчанию

```
public Car() {   }
```

Если создан хотя бы один конструктор в классе, конструктор по умолчанию НЕ СОЗДАЁТСЯ

Конструктор **не может** быть static, abstract или final

Перегрузка конструкторов

Конструктор, как и любой другой метод можно перегружать. При создании объекта будет вызван конструктор в зависимости от входящих параметров.

```
public Animal() {
    this.age = 14;
    this.height = 60;
}

public Animal(int age) {
    this.age = age;
    this.height = 66;
}

public Animal(int age, int height) {
    this.age = age;
    this.height = height;
}
```

```
Animal animal = new Animal();
```

```
Animal animal = new Animal(15);
```

```
Animal animal = new Animal(20, 180);
```

Ключевые слова

Статический контекст

Модификатор static — определяет поле или метод, который принадлежит непосредственно КЛАССУ. Для вызова статического поля или метода не требуется создавать экземпляр класса.

Вызов **не** статического метода

```
class A {
    public int method(){…};
}
```

```
public static void main(String[] args ) {
    A a = new A();
    a.method();
}
```

Вызов **статического** метода

```
class A {
    static int method() {…};
}
```

```
public static void main(String[] args ) {
    A.method();
}
```

Q&A