**Оператор instanceof**

[Java Core](https://javarush.ru/quests/QUEST_JAVA_CORE)

[Уровень 4](https://javarush.ru/quests/lectures?quest=QUEST_JAVA_CORE&level=4), Лекция 1

— Привет, Амиго! Ты уже раньше знакомился с оператором **instanceof**. Сегодня я расскажу тебе, как и где его можно использовать. **instanceof** – это очень простой и эффективный в использовании оператор.

— Звучит, как реклама!

— Он, и правда, очень простой. Он используется в виде: **«объект» instanceof «класс».**

Он проверяет, является ли объект объектом определенного класса. Все еще проще, чем я говорю. Смотри пример:

|  |  |
| --- | --- |
| **Код** | **Описание** |
| Object o = new Integer(3);  boolean isInt = o instanceof Integer; | **isInt** будет равно **true**. Объект, на который ссылается переменная o, является объектом класса **Integer**. |
| Object o = "Mama";  boolean isInt = o instanceof Integer; | **isInt** будет равно **false**. Объект, на который ссылается переменная o, не является объектом класса **Integer**, он является объектом класса **String**. |
| InputStream is = new FileInputStream("");  boolean isFIS = is instanceof  FileInputStream; | **isFIS** будет равно **true**. Объект, на который ссылается переменная is, является объектом класса **FileInputStream**. |

4

Задача

Java Core,  4 уровень,  1 лекция

Набираем код Ӏ Java Core: 4 уровень, 1 лекция

Java Core: 4 уровень, 1 лекция. Иногда думать не надо, строчить надо! Как ни парадоксально звучит, порой пальцы «запоминают» лучше, чем сознание. Вот почему во время обучения в секретном центре JavaRush вы иногда встречаете задания на набор кода. Набирая код, вы привыкаете к синтаксису и зарабатываете немного материи. А ещё — боретесь с ленью.

— Да, очень просто.

— Этот оператор учитывает и наследование. Вот смотри.

|  |  |
| --- | --- |
| **Код** | **Описание** |
| class Animal  {  }  class Cat extends Animal  {  }  class Tiger extends Cat  {  } | Тут мы видим три объявленных класса: животное, кот и тигр. Кот наследуется от Животного. А Тигр от Кота. |
| Object o = new Tiger();  boolean isCat = o instanceof Cat;  boolean isTiger = o instanceof Tiger;  boolean isAnimal = o instanceof Animal; | **isCat** будет равно **true**. **isTiger** будет равно **true**. **isAnimal** будет равно **true**. |
| Object o = new Animal();  boolean isCat = o instanceof Cat;  boolean isTiger = o instanceof Tiger;  boolean isAnimal = o instanceof Animal; | **isCat** будет равно **false**. **isTiger** будет равно **false**. **isAnimal** будет равно **true**. |

И даже интерфейсы:

|  |  |
| --- | --- |
| **Код** | **Описание** |
| interface Moveable  {  }  class Cat  {  }  class TomCat extends Cat implements Moveable  {  } | Создадим два класса: Cat, TomCat и интерфейс Moveable |
| Cat o = new TomCat();  boolean isCat = o instanceof Cat;  boolean isMoveable = o instanceof Moveable;  boolean isTom = o instanceof TomCat; | **isCat** будет равно **true**. **isMoveable** будет равно **true**. **isTom** будет равно **true**. |
| Cat o = new Cat();  boolean isCat = o instanceof Cat;  boolean isMoveable = o instanceof Moveable;  boolean isTom = o instanceof TomCat; | **isCat** будет равно **true**. **isMoveable** будет равно **false**. **isTom** будет равно **false**. |

Оператор instanceof имеет вид: **a** **instanceof** **B**.

Другими словами, оператор **instanceof** вернет значение **true**, если:

**1)** переменная **а** хранит ссылку на объект типа **B**

**2)** переменная **a** хранит ссылку на объект, класс которого унаследован от **B**

**3)** переменная **а** хранит ссылку на объект реализующий интерфейс **B**

Иначе оператор **instanceof** вернет значение **false**.

— Понятно. А зачем это нужно, дядя Риша?

— Об этом сегодня тебе расскажет Элли. Это очень хороший оператор. Сегодня ты в этом убедишься.