Theory: The Object class

7 minutes 0 / 5 problems solved

§1. The root class in Java

The Java Standard Library has a class named `Object` that is the parent of all standard classes and your custom classes by default. Every class extends this class implicitly, therefore it's a root of inheritance in Java programs. The class belongs to the `java.lang` package that is imported by default.

Let's create an instance of the `Object`.

```
Object anObject = new Object();
```

The `Object` class can refer to an instance of any class because any instance is a kind of `Object` (*upcasting*).

```
Long number = 1_000_000L;
Object obj1 = number; // an instance of Long can be cast to Object

String str = "str";
Object obj2 = str; // the same with the instance of String
```

When we declare a class, we can explicitly extend the `Object` class. However, there is no point, since the extension is already done implicitly. We advise you to not make your code redundant, but here's an example, just in case:

```
class A extends Object { }
```

In your own solutions, it is enough to write `class A { }` instead of this.

§2. Methods provided by the Object class

The `Object` class provides some common methods to all subclasses. It has nine instance methods (excluding overloaded methods) which can be divided into four groups:

- *threads synchronization*: `wait`, `notify`, `notifyAll`;
- *object identity*: `hashCode`, `equals`;
- *object management*: `finalize`, `clone`, `getClass`;
- *human-readable representation*: `toString`;

This way of grouping methods isn't supposed to be perfect, but it can help you remember them. Here's a more detailed explanation of the methods:

- The first group of methods (`wait`, `notify`, `notifyAll`) are for working in multithreaded applications.
- The method `hashCode` returns a hash code value for the object.
- The method `equals` indicates whether some other object is "**equal to**" this particular one.
- The method `finalize` is called by the garbage collector (GC) on an object when the GC wants to clean it up. (**Note:** this method has been deprecated as of JDK 9).
- The method `clone` creates and returns a copy of the object.
- The method `getClass` returns an instance of `Class`, which has information about the runtime class.
- The method `toString` returns a string representation of the object.

Some of the methods listed above are native which means they are implemented in the "**native**" code. It's typically written in C or C++. Native methods are usually used to interface with system calls or libraries written in other programming languages.

In the following topics, we will consider methods of the class in more detail.

Current topic:

The Object class

Topic depends on:

✕ Referencing subclass objects

Topic is required for:

toString()

Generics and Object

Serialization basics

hashCode() and equals()

Table of contents:

↑ The Object class

§1. The root class in Java

§2. Methods provided by the Object class

Feedback & Comments

295 users liked this theory. **2** didn't like it. What about you?

Start practicing

Comments (11) Hints (0) Useful links (0) Show discussion