Theory: The main method

7 minutes 5 / 12 problems solved

Start practicing

§1. The declaration of the main method

Java is primarily an object-oriented language. It means a Java program can be considered as a collection of objects that communicate via calling each other's methods. A typical Java program includes a lot of classes, interfaces, objects and other concepts from the object-oriented programming.

Even the simplest "procedural-style" program should have at least one class and the main method inside to start the program. The main method is the entry point for any application. Ever since Java 7 there has been no other way to start an application without this method (excluding the case when you start your application inside a special container for applications but it is not considered in our materials).

Let's see an example of the simplest application that prints the text "Hello, Java" in the standard output:

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello, Java");
    }
}
```

Here is a class named `Main`. The class contains **the main method** for starting the program.

It is important to mention that a class containing the main method can have any name, but the main method should always have the name `main`.

Let's take a closer look at the declaration of the main method:

```
public static void main(String[] args)
```

- the keyword `public` indicates that the method can be invoked from everywhere;
- the keyword `static` indicates the method can be invoked without creating an instance of the class;
- the keyword `void` indicates the method doesn't return any value;
- the array variable `args` contains arguments entered at the command line, the array is empty if there are no arguments.

As you can see, even the simplest Java application contains a lot of concepts. All of them will be studied in the next topics related to methods and the object-oriented programming. Now you should just understand how to write and run a simple java program with **the main method**.

§2. Invalid declarations of the main method

If the main method has an invalid declaration, two cases are possible:

- your program cannot be compiled
- your program is successfully compiled but can't be started

Your program cannot be compiled. It is a case when the main method declaration breaks the syntax of Java.

Examples:

- invalid method declaration: no returning value (even `void`).

```
public static main(String[] args)
```

Current topic:

✓ The main method — Stage 3

Topic depends on:

✓ Declaring a method — Stage 3

Topic is required for:

✓ Functional decomposition — Stage 3

Command-line arguments

Table of contents:

- invalid method declaration: a mistake in the keyword (`pubic` instead of `public`).

```
pubic static void main(String[] args)
```

A program can be compiled but cannot be run. It is a case when the main method has a correct declaration as a regular method but doesn't satisfy the specific requirement of the main method.

Examples:

- invalid arguments (should be `String[] args`)

```
public static void main(String args) {
    System.out.println("Hello, Java");
}
```

- the method declaration has no keyword `static`

```
public void main(String[] args) {
    System.out.println("Hello, Java");
}
```

In both cases, an error happens at runtime.

§3. Conclusion

So, the main method is the entry point of any Java program. It has a very specific syntax which you need to remember.

Report a typo

474 users liked this theory. **3** didn't like it. **What about you?**

Start practicing

Comments (11) Hints (0) Useful links (1) Show discussion