Theory: Comments

8 minutes 6 / 13 problems solved

Start practicing

Inside a Java program, you can write a special text that will be ignored by the java compiler — known as the **comment**. Comments allow you to exclude code from the compilation process (disable it) or clarify a piece of code to yourself or other developers. In our materials, we use comments in the theory and practice lessons to explain how and why our code works.

The Java programming language supports three kinds of comments.

§1. End-of-line comments

The java compiler ignores any text from `//` to the end of the line.

```
class Program {
    public static void main(String[] args) {
        // The line below will be ignored
        // System.out.println("Hello, World");
        // It prints the string "Hello, Java"
        System.out.println("Hello, Java"); // Here can be any comment
    }
}
```

In the example above the text after `//` is ignored by the compiler.

§2. Multi-line comments

The compiler ignores any text from `/*` and the nearest `*/`. It can be used as multiple and single-line comments.

```
class Program {
    public static void main(String[] args) {
        /* This is a single-line comment */
        /*  This is an example of
            a multi-line comment */
  }
}
```

You can use comments inside other comments:

```
class Program {
    public static void main(String[] args) {
        /*
        System.out.println("Hello"); // print "Hello"
        System.out.println("Java");  // print "Java"
        */
    }
}
```

The part of the code above is ignored by the compiler because of `/* ... */` comments.

§3. Java documentation comments

The compiler ignores any text from `/**` to `*/` just like it ignores multi-line comments.

These kinds of comments can be used to automatically generate documentation about your source code by using the **javadoc** tool. Usually, these comments are placed above declarations of classes, interfaces, methods and so on. Some special labels such as `@param` or `@return` are often used for controlling the tool. However, they are optional and we will not deal with them for now. Just don't be surprised in case you see one.

See the example below.

Current topic:

✓ Comments — Stage 1

Topic depends on:

✓ Printing data — Stage 1

Topic is required for:

✓ Scanning the input — Stage 1

Table of contents:

```
class Program {
    /**
     * The main method accepts an array of string arguments
     *
     * @param args from the command line
     */
    public static void main(String[] args) {
        // do nothing
    }

}
```

Do not be afraid if you have not understood **the documentation comments** completely. This will be considered in other topics.

Report a typo

1638 users liked this theory. **11** didn't like it. **What about you?**

Start practicing

Comments (2) Hints (0) Useful links (0) Show discussion