Theory: Types and variables

7 minutes 5 / 12 problems solved

Start practicing

27343 users solved this topic. Latest completion was 17 minutes ago.

§1. Declaring and initializing

A **variable** is a placeholder for storing a value of a particular **type**: a string, a number, or something else. Every variable has a **name** (also known as an **identifier**) to distinguish it from others. Before you start using a variable, you must declare it.

The general form of declaration is the following:

```
DataType variableName = initialization;
```

The left part of this statement describes the variable, and the right part describes something that is assigned to it.

- The **type** (or **data type**) of a variable determines what possible operations can be performed on the variable and which values can be stored in it. Here we use a non-existing data type (**DataType**) to demonstrate the general form of declaration.
- The **name** (or **identifier**) distinguishes the variable from others. The name of a variable cannot start with a digit; it usually starts with a letter. Always try to choose meaningful and readable names for variables to make your code easy to understand.
- The **assignment operator** denoted as `=` is used to assign a single value or a result of an expression to a variable.
- The **initialization** is a value or a result of an expression that is assigned to the variable.

According to this declaration, we can declare a variable of the type **String** and assign the word **"java"** to it:

```
String language = "java";
```

We can also declare a variable of the type `int` to store an integer number:

```
int numberOfApples = 5;
```

The case in the name of a variable makes a difference: `language` is not the same as `Language`.

Variables can store not only strings and numbers, but also characters and other data types which we will learn about later in the next topics.

§2. Accessing the value of a variable

Once a variable has been declared, its value can be accessed and modified using the name. In the example below, we declare a variable and then print it:

```
String dayOfWeek = "Monday";
System.out.println(dayOfWeek); // Monday
```

It is also possible to assign a value of one variable to another one:

```
int one = 1;
int num = one;
System.out.println(num); // 1
```

One important feature of variables is that they can be changed. You don't need to declare a variable again to change its value; just assign a new value to it using the `=` operator.

Let's declare a variable named `dayOfWeek` and print its value before and after changing:

Current topic:

✓ Types and variables Stage 1

Topic depends on:

✓ Printing data Stage 1

Topic is required for:

✓ Sizes and ranges Stage 1

✓ Naming variables Stage 1

✓ Scanning the input Stage 1

✓ Bitwise and bit-shift operations

✓ Boolean and logical operations Stage 2

Primitive and reference types Stage 2

Table of contents:

```
String dayOfWeek = "Monday";
System.out.println(dayOfWeek); // Monday

dayOfWeek = "Tuesday";
System.out.println(dayOfWeek); // Tuesday
```

There is one restriction for variables: you can only assign a value of the same type as the type of the initial variable. So, the following code is not correct:

```
int number = 10;
number = 11; // ok
number = "twelve"; // it does not work!
```

§3. Alternative forms of declaration

There are several alternative forms of declaration which are less commonly used in practice. Here are several of them in particular examples.

- Declaring several variables of the same type as a single statement:

```
String language = "java", version = "8 or newer";
```

- Separating declaration and initialization into statements:

```
int age; // declaration
age = 35; // initialization
```

However, as we have already noted, these forms are rarely used.

§4. Type inference

Since Java 10, you can write **var** instead of a specific type to force automatic type inference based on the type of assigned value:

```
var variableName = initialization;
```

Here are two examples below:

```
var language = "Java"; // String
var version = 10; // int
```

This feature can be a bit controversial: on the one hand, it allows your code to be more concise. On the other hand, since it doesn't indicate the type explicitly, it may affect the code readability in a bad way. For now, it's enough to understand the basic idea. We will not use type inference in our theory so that our educational platform is suitable for people who use earlier versions of Java. But if you would like to practice it, you may use type inference in our exercises as they fully support Java 10.

Report a typo

Thanks for your feedback!

Start practicing

This content was created over 3 years ago and updated 8 days ago. Share your feedback below in comments to help us improve it!

Comments (36) Hints (0) Useful links (0) Show discussion