


```
1 double one = 1.0;
2 double number = one + 1.5; // 2.5
3
4 double a = 1.75;
5 double b = 5.0;
6 double c = b - a; // 3.25
7
8 double pi = 3.1415;
9 double squaredPi = pi * pi; // 9.86902225
```

For `double` and `float` operands, the operator `/` performs a division, not an integer division.

```
1 System.out.println(squaredPi / 2); // 4.934511125
```

Pay attention to an important thing that beginners often miss:

```
1 double d1 = 5 / 4; // 1.0
2 double d2 = 5.0 / 4; // 1.25
```

In the first case, we perform integer division that produces an integer result and then assign the result to `d1`. In the second case, we perform a real division that produces double value and then assign the value to `d2`.

§3. Errors during computations

Be careful! Operations with floating-point numbers can produce an inaccurate result:

```
1 System.out.println(3.3 / 3); // prints 1.0999999999999999
```

Errors can accumulate during computation. In the following example we calculate the sum of ten decimals `0.1`:

```
1 double d = 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1;
2 System.out.println(d); // it prints 0.9999999999999999
```

In the following lessons, we will find out how to deal with this issue. For now, just take it into consideration.

If you want to learn more about floating-point numbers, you may [read this article](#) as an addition.

§4. Reading floating-point numbers

You can use `Scanner` to read the values of both of these types from the standard input.

```
1 Scanner scanner = new Scanner(System.in);
2
3 float f = scanner.nextFloat();
4 double d = scanner.nextDouble();
```

We recommend you use `double` to solve our programming problems.

As an example, consider a program that calculates the `area` of a triangle. To find it, the program reads the `base` and the `height` from the standard input, then multiplies them, and divides by 2. Note that the base and the height are perpendicular to each other.

```
1  import java.util.Scanner;
2
3  public class AreaOfTriangle {
4      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          double base = scanner.nextDouble();
8          double height = scanner.nextDouble();
9
10         double area = (base * height) / 2;
11
12         System.out.println(area);
13     }
14 }
```

Let's calculate the area of a triangle with a base of 3.3 meters and a height of 4.5 meters.

Input 1:

```
1  | 3.3 4.5
```

Output 1:

```
1  | 7.425
```

As you can see, it's area is 7.425 square meters!

Keep in mind that the output of this program may have a lot of zeros like the output below because an operation with floating-point numbers can produce inaccurate results.

Input 2:

```
1  | 2.2 4.01
```

Output 2:

```
1  | 4.4110000000000005
```

It is possible to round or format a double result, but we will not do it in this lesson. In the coding problems, output a result as-is.

§5. The decimal separator

If you solve our coding problems locally or try to repeat our examples, you may encounter a problem with your computer having different locale settings. In this case, the `Scanner` cannot read floating-point numbers with the dot character (3.1415). Try to input numbers written with the comma separator (3,1415).

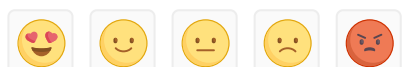
If you want to use the dot character without modifying your local settings, try using the following code to create a scanner:

```
1  | Scanner scanner = new Scanner(System.in).useLocale(Locale.US);
```

You do not need to change anything when submitting your solutions in our forms.

[Report a typo](#)

423 users liked this theory. 10 didn't like it. What about you?



Start practicing

[Comments \(8\)](#)

[Hints \(0\)](#)

[Useful links \(0\)](#)

[Show discussion](#)