Laboratory work #1 | Programming basic concepts

Lab objectives

As a result of this laboratory work you will know basics of programming in Java:

- using of Eclipse integrated development environment (IDE);
- using of standard function to do simple calculations;
- using of primitive data types, variables.
- doing of data input and output with console (terminal).

Overview

In this lab you will implement complicated mathematical formula. To do it you will use default trigonometrical functions like sin, cos, tg, ctg. Any program is useless if it doesn't provide results, and to do that you will implement console output. To get the data from user you will implement console input.

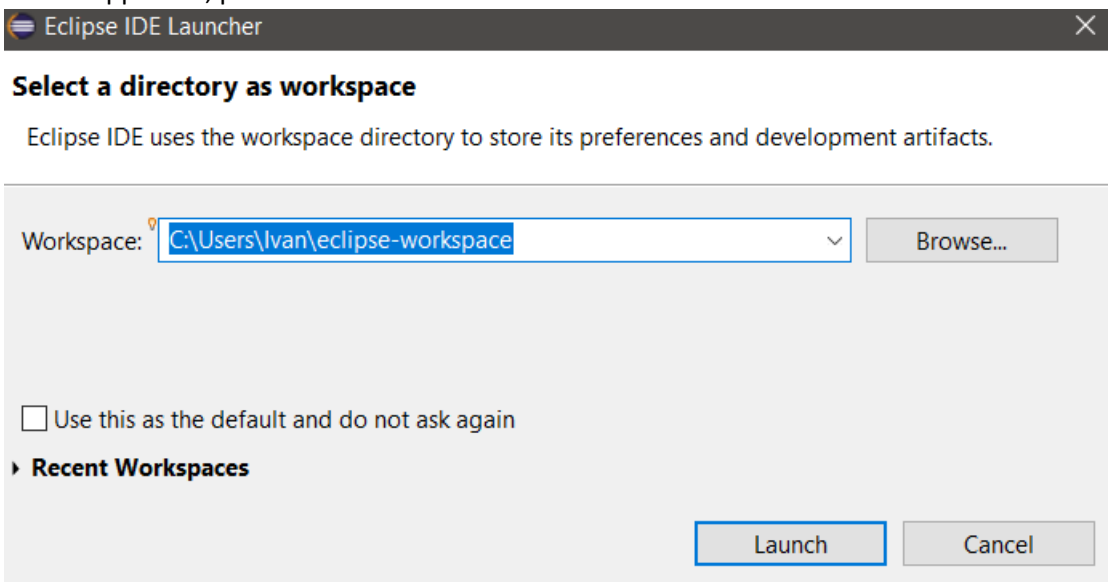
Task

1. Run the Eclipse integrated development environment (IDE). Check that the environment works fine and write your first program.
2. Get acquainted with your variant. The variants you can find at the end of this document.
3. Implement the simple application that will do calculations by formula (in your variant).
4. Add the input and output dialogues to make interaction with a user possible.
5. Refactor your code – organize parts of your code in methods.
6. Check the calculation results with results on a piece of paper on in the other calculator.

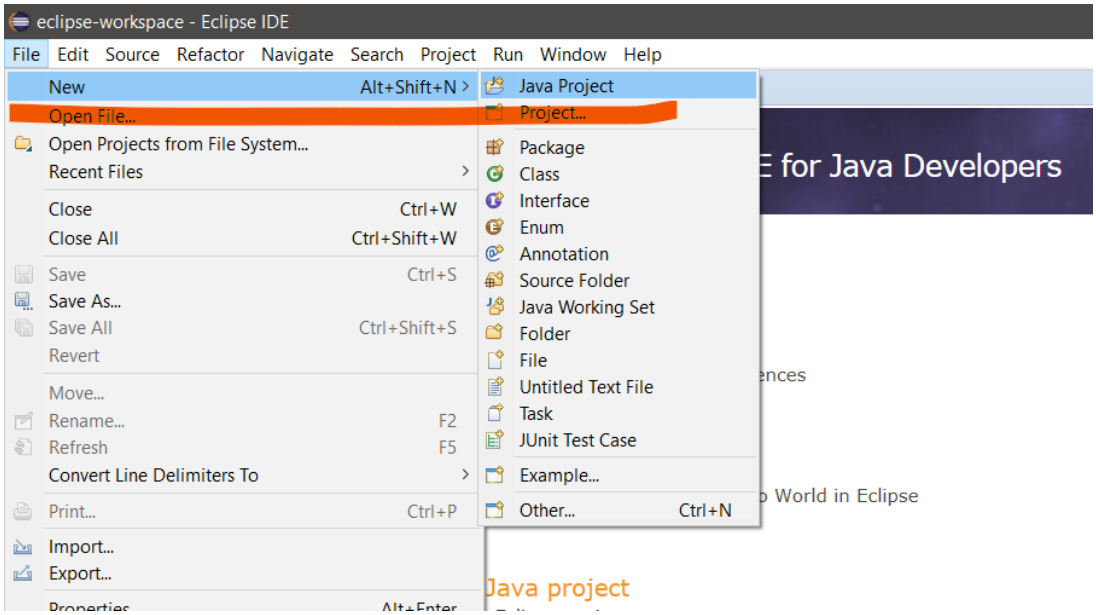
Instructions

1. Run the Eclipse integrated development environment (IDE). Create the new Java Project.

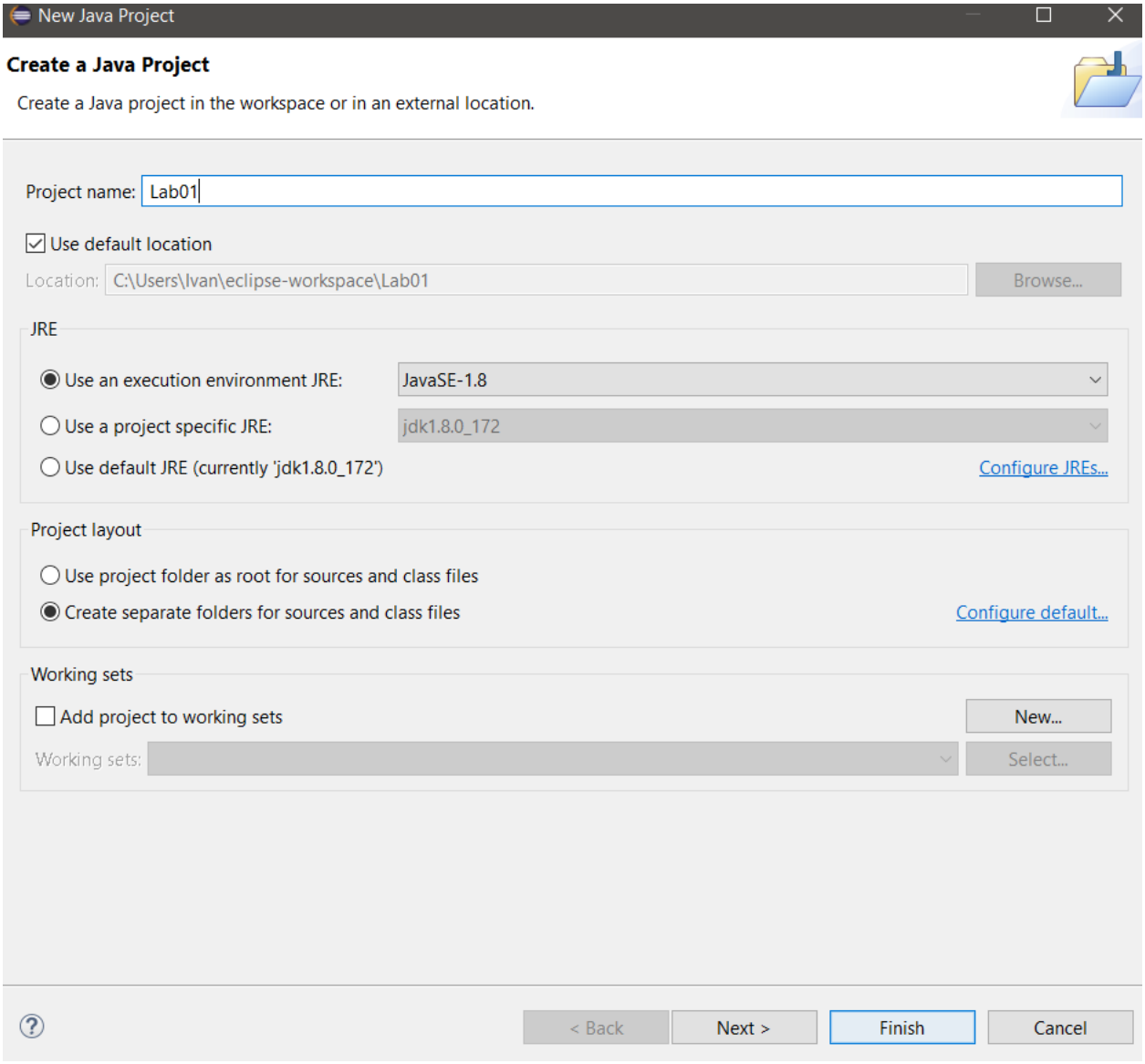
1.1. If this window appeared, press the "Launch" button.



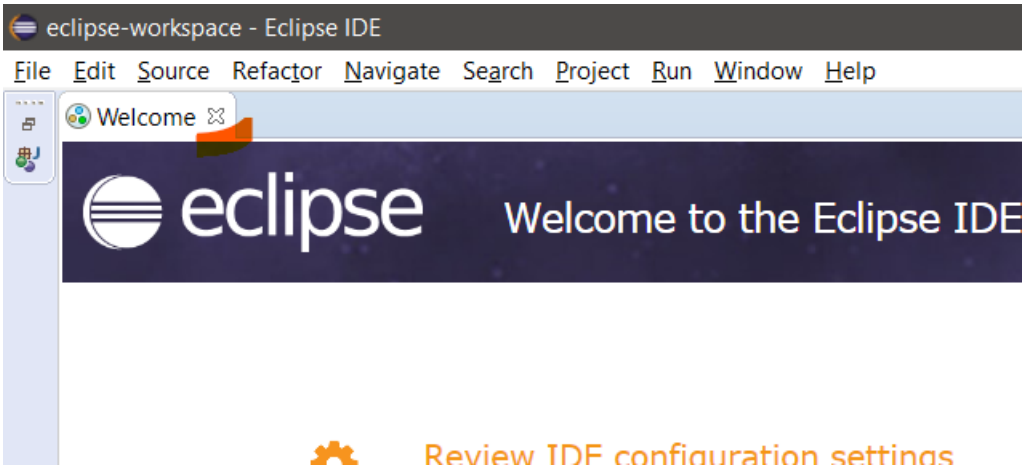
1.2. Choose the “File / New / Java Project” menu item.



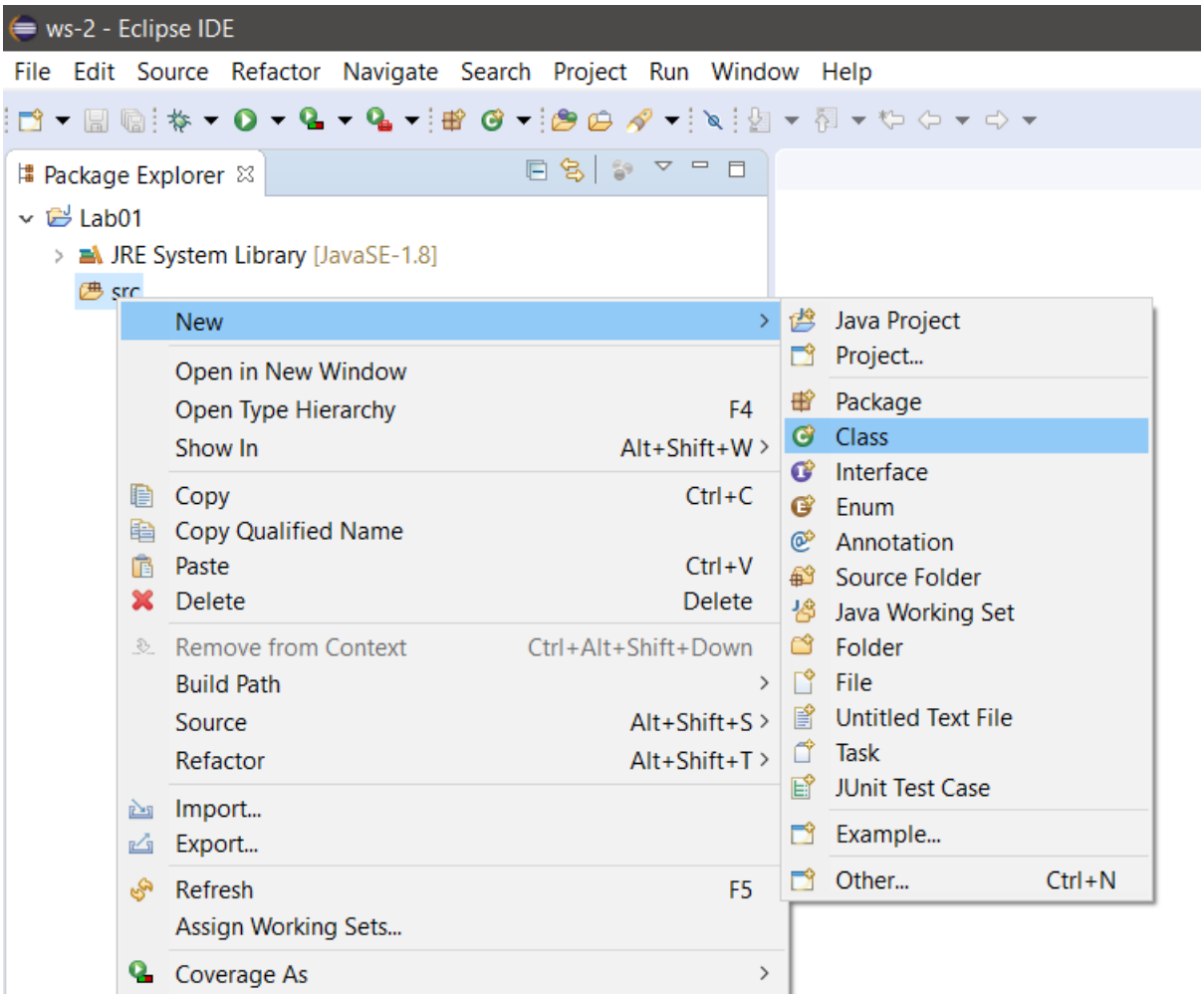
1.3. Write the name of a project – for example, Lab01, and press “Finish” button.



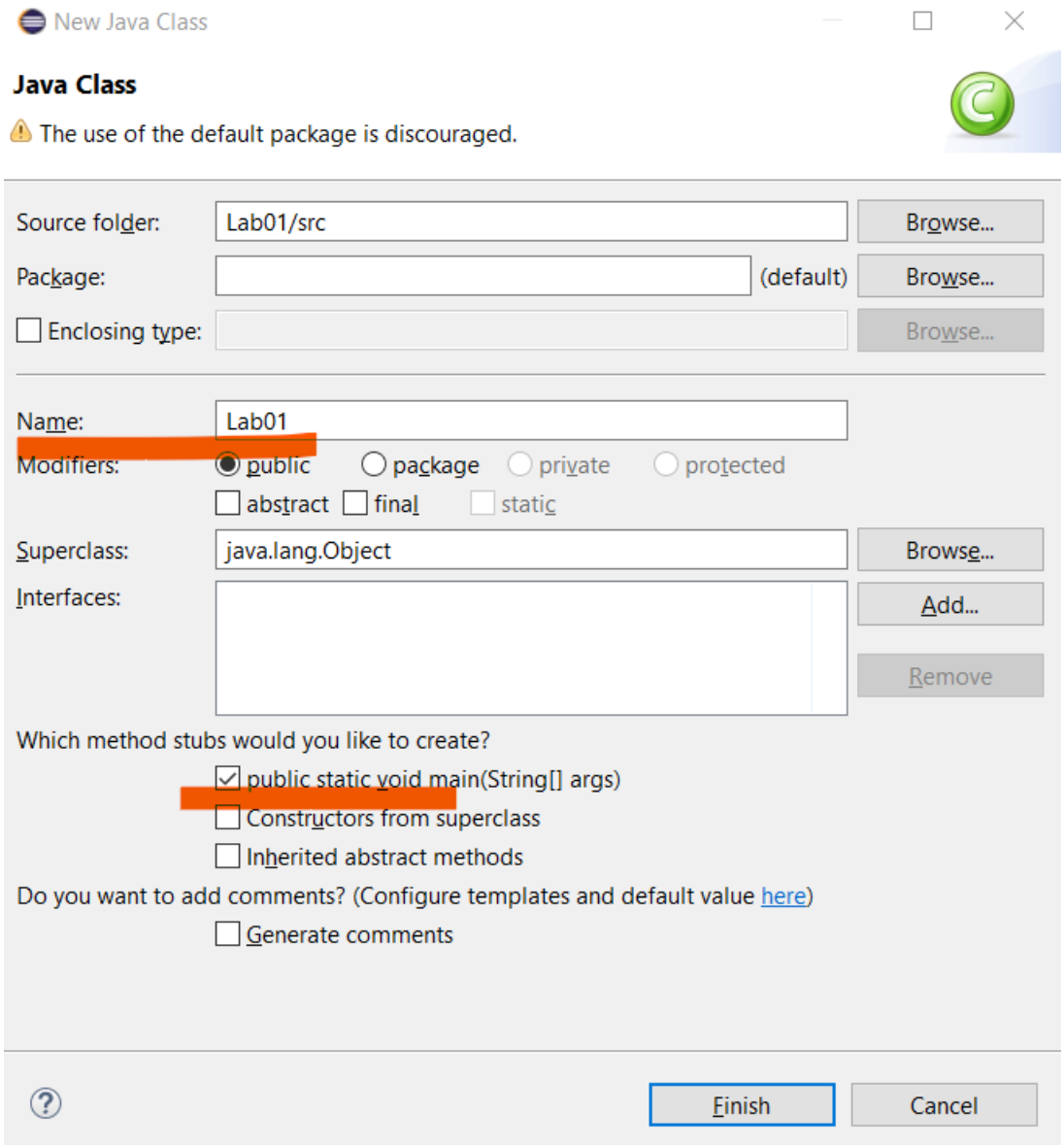
1.4. You will see your workspace (as shown in the screenshot below). If you can't see this, close the Welcome-page (press the close button near in tab, see the next screenshot).



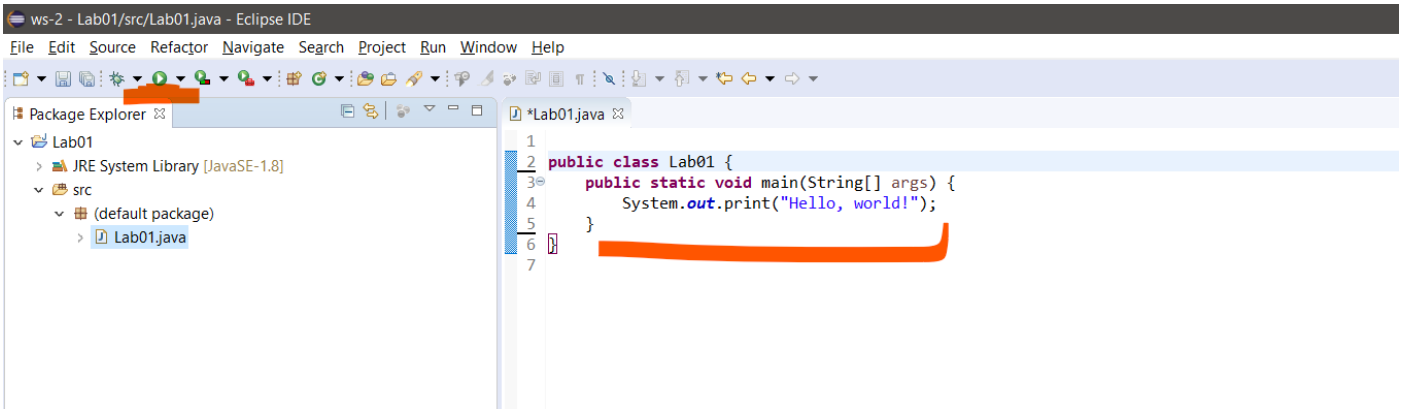
1.5. Right-click on the folder "src" as shown below. Select "New / Class" menu item.



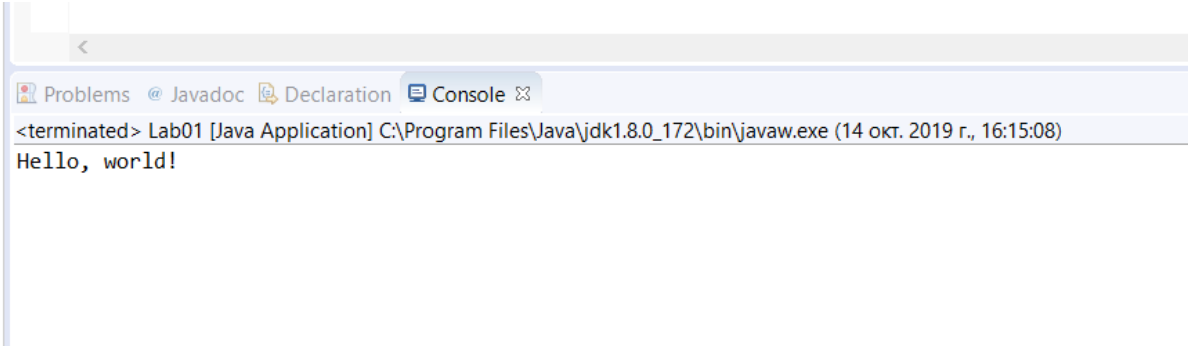
1.6. Enter the name of the class and check those checkboxes in state like at the screenshot below. Press the “Finish” button.



1.7. In the opened window write your first program and build it with the green “Run” button (see the screenshot).



1.8. See the results. If some dialogues will appear, press the “Ok” button or “Apply” button.



2. Get acquainted with your variant. The variants you can find at the end of this document.

2.1. For example, your variant defines the formulas listed below.

$$z_1 = \frac{(m-1)\sqrt{m} - (n-1)\sqrt{n}}{\sqrt{m^3 n} + nm + m^2 - m}$$

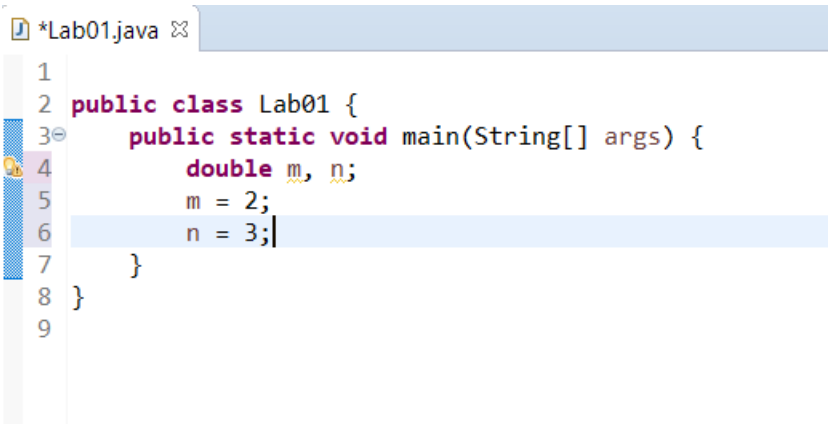
$$z_2 = \frac{\sqrt{m} - \sqrt{n}}{m}$$

It means that you should calculate two values: z1 and z2, with m and n values stored in variables, or will be given by the user (point №6).

Write this formula on a piece of paper, write ignored operations, like multiplication – in mathematical notation it is possible to ignore it, but in programming languages, you can’t ignore no one operation.

3. Implement the simple application that will do calculations by formula (in your variant).

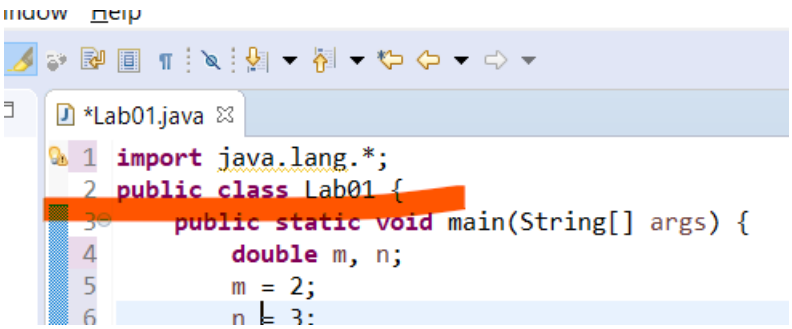
3.1. Declare variables that are used inside formula – in this variant there are m and n, in other cases it could be a, b, alpha, beta, etc. Assign the default values for these variables (2 and 3 for example).



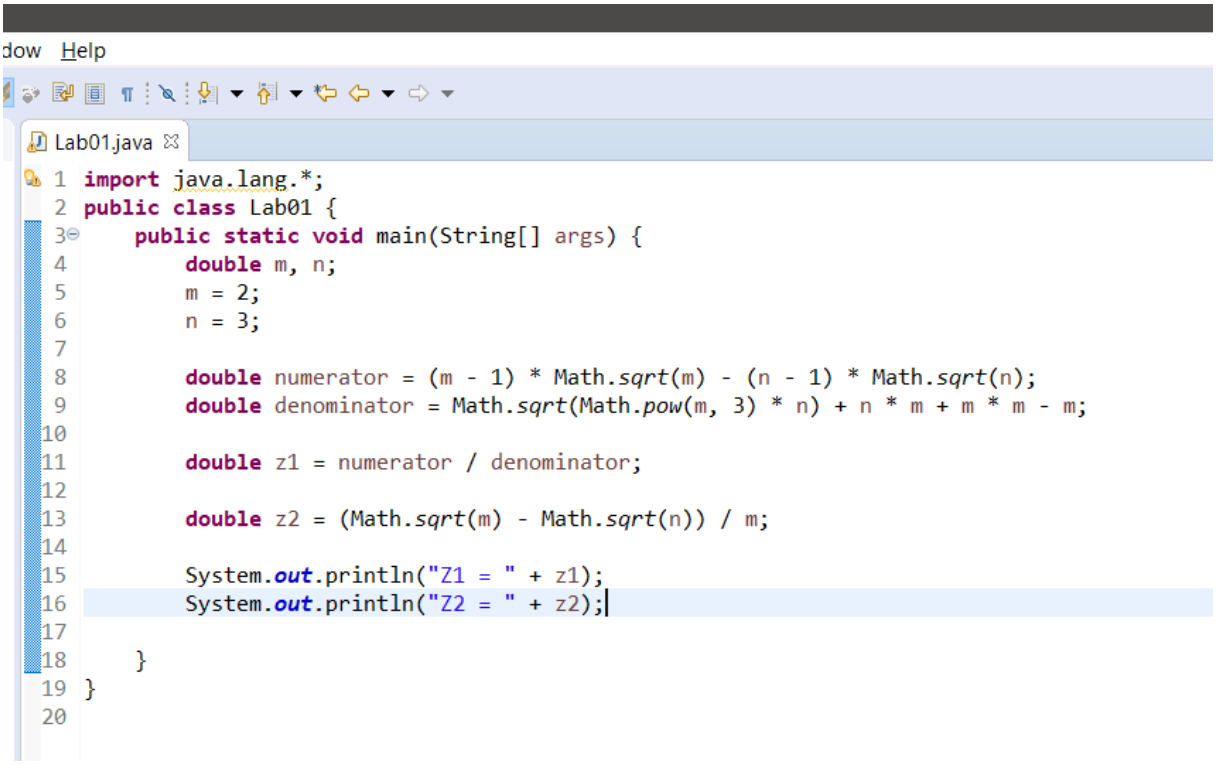
3.2. Write the formulas in Java.

To use mathematical function like **sin, cos, tg, ctg, number Pi, get square root, etc,** you should write the first line (***before the public class Lab01 line!***) as shown below:

import java.lang.*;



Here you can see the resulting program:

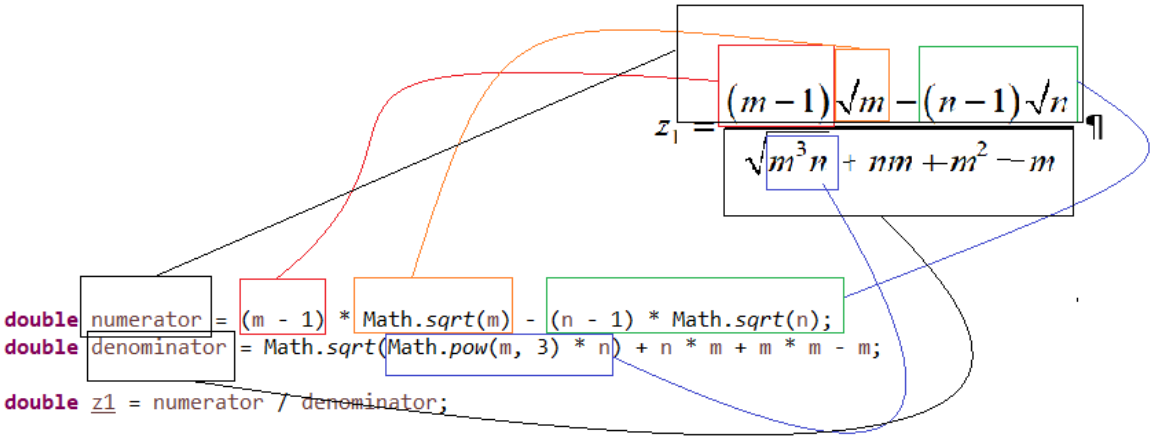


The source formula is too complicated to write it in the single line. As a result, it was written in two parts: the first part – numerator, the second part – denominator. The second formula no so complicated, it is short, and it is possible to write it in the single line.

You can write all formula in a single line, but it isn't good to read. In this case don't remember the brackets around all expression – numerator, denominator, like:

((m – 1) * Math.sqrt(m) – (n – 1) * Math.sqrt(n))

Here you can see matching between the parts of a formula in Java and in mathematical notation:



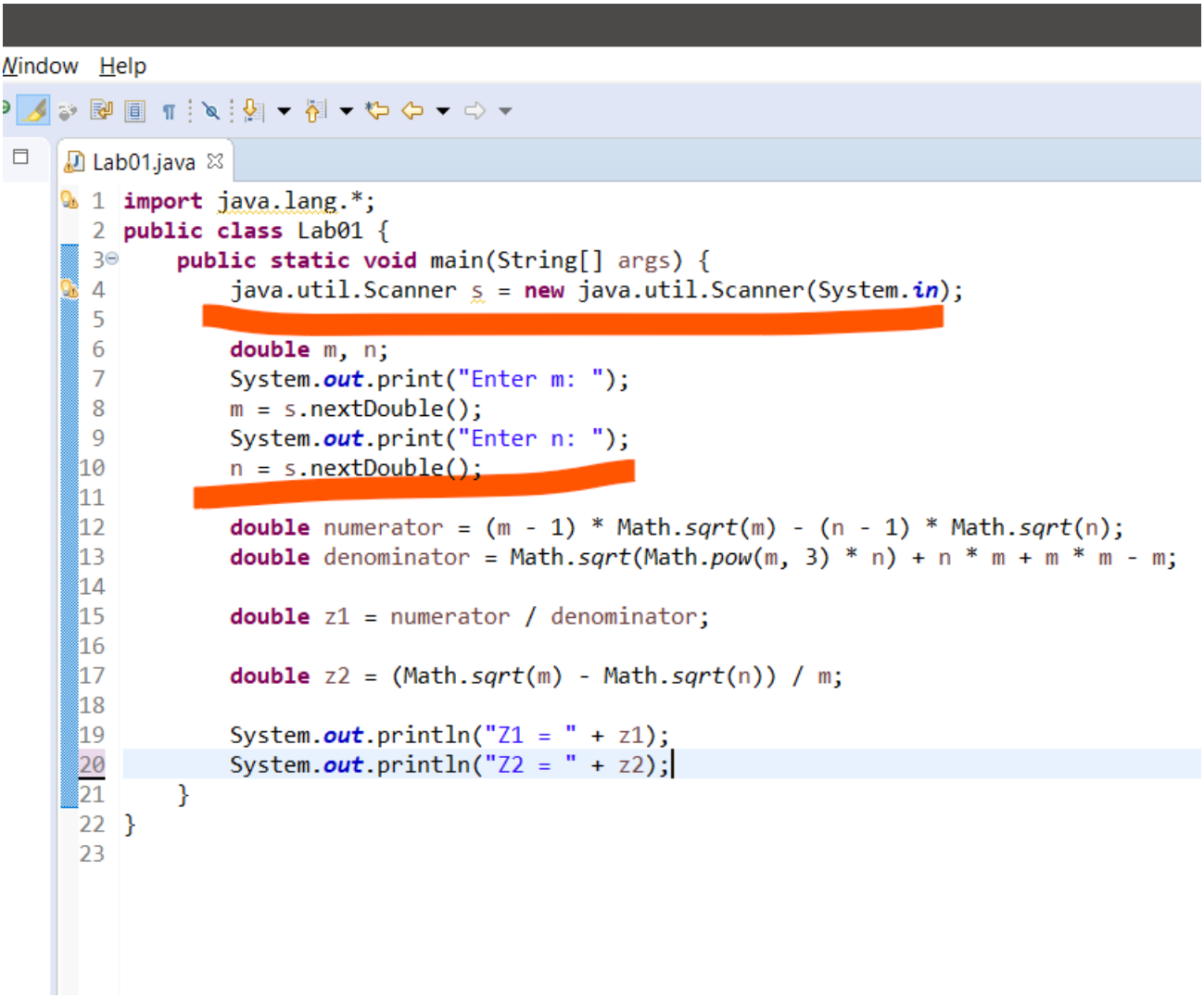
Here listed the operations and example of its' usage in Java.

Name of the operation	**Mathematical operation**	**Java code**
Multiplication	(a – b)c	(a – b) * c
Division	$\frac{x}{y}$	X / y
More complex division	$\frac{x-5}{y+4}$	(x – 5) / (y + 4)
Square	x^2	(x * x)
Power of N	x^3	Math.pow(x, 3)
Square root	$\sqrt{x}$	Math.sqrt(x)
Sin	sin(x)	Math.sin(x)
Cos	cos(x)	Math.cos(x)
Tg (Tan)	tg(x)	Math.tan(x)
Ctg (Cotan)	ctg(x)	1 / Math.tan(x)
Note: for trigonometrical functions the argument in radians degrees!		

4. Add the input and output dialogs to make interaction with a user possible

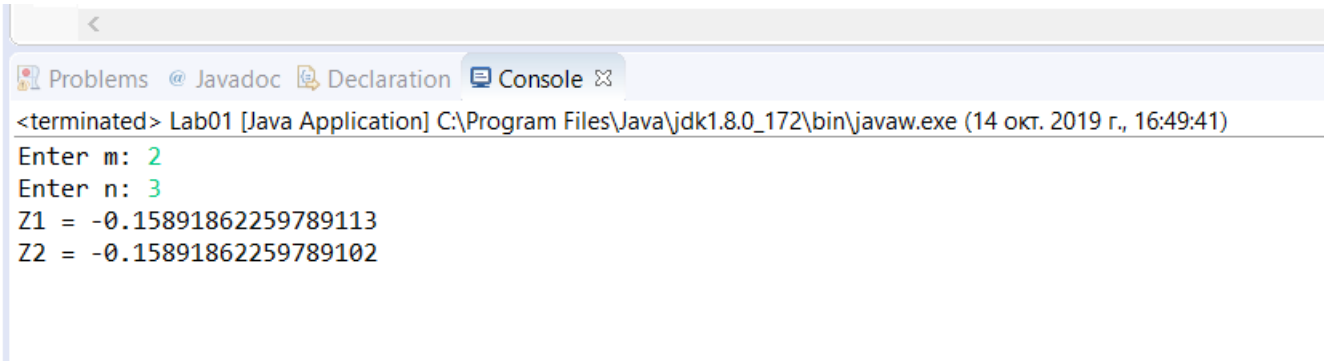
To make possible read data from the user you should add code that makes console input.

To do that you should write this lines:



Be sure, that the order of added lines exactly like shown on screenshot!

Now, when you will run your program, it will ask you to enter m and n, you can input values and press Enter after each number.



5. **Refactor your code – let's make it more readable**. When you write nice code, which is easy to read and understand, you will not make mistakes, bugs!
   5.1. Let's create methods:
      5.1.1.calcZ1 will do computations to get Z1 value.
      5.1.2.calcZ2 will do computations to get Z2 value.
      5.1.3.getValue will read double value from the console.
      5.1.4.Method main will use all this methods.
   5.2. The full source code of example:

```
import java.lang.*;
public class Lab01 {
	public static double readValue(String message) {
		java.util.Scanner s = new java.util.Scanner(System.in);

		System.out.print(message);

		return s.nextDouble();
	}

	public static double calcZ1(double m, double n) {
		double numerator = (m - 1) * Math.sqrt(m) - (n - 1) * Math.sqrt(n);
		double denominator = Math.sqrt(Math.pow(m, 3) * n) + n * m + m * m - m;

		double z1 = numerator / denominator;

		return z1;
	}

	public static double calcZ2(double m, double n) {
		double z2 = (Math.sqrt(m) - Math.sqrt(n)) / m;
		return z2;
	}

	public static void main(String[] args) {
		double m = readValue("Enter m: ");
		double n = readValue("Enter n: ");

		double z1 = calcZ1(m, n);
		double z2 = calcZ2(m, n);

		System.out.println("Z1 = " + z1);
		System.out.println("Z2 = " + z2);
	}
 }
```

Variants

Variant 1

$$z_1 = 2\sin^2(3\pi - 2\alpha)\cos^2(5\pi + 2\alpha)$$

$$z_2 = \frac{1}{4} - \frac{1}{4}\sin\left(\frac{5}{2}\pi - 8\alpha\right)$$

Variant 2

$$z_1 = \cos\alpha + \sin\alpha + \cos 3\alpha + \sin 3\alpha$$

$$z_2 = 2\sqrt{2}\cos\alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right)$$

Variant 3

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos\alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin\alpha$$

Variant 4

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos\alpha - \cos 3\alpha + \cos 5\alpha}$$

$$z_2 = \mathrm{tg}3\alpha$$

Variant 5

$$z_1 = 1 - \frac{1}{4}\sin^2 2\alpha + \cos 2\alpha$$

$$z_2 = \cos^2\alpha + \cos^4\alpha$$

Variant 6

$$z_1 = \cos\alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$$

$$z_2 = 4\cos\frac{\alpha}{2} \cdot \cos\frac{5}{2}\alpha \cdot \cos 4\alpha$$

Variant 7

$$z_1 = \cos^2\left(\frac{3}{8}\pi - \frac{\alpha}{4}\right) - \cos^2\left(\frac{11}{8}\pi + \frac{\alpha}{4}\right)$$

$$z_2 = \frac{\sqrt{2}}{2}\sin\frac{\alpha}{2}$$

Variant 8

$$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4}\sin^2 2x - 1$$

$$z_2 = \sin(y + x) \cdot \sin(y - x)$$

Variant 9

$$z_1 = (\cos\alpha - \cos\beta)^2 - (\sin\alpha - \sin\beta)^2$$

$$z_2 = -4\sin^2\frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta)$$

Variant 10

$$z_1 = \frac{\sin\left(\frac{\pi}{2} + 3\alpha\right)}{1 - \sin(3\alpha - \pi)}$$

$$z_2 = \mathrm{ctg}\left(\frac{5}{4}\pi + \frac{3}{2}\alpha\right)$$