

INTRODUCTION TO JAVA

Java 1.0



VARIABLES

Lesson # 02



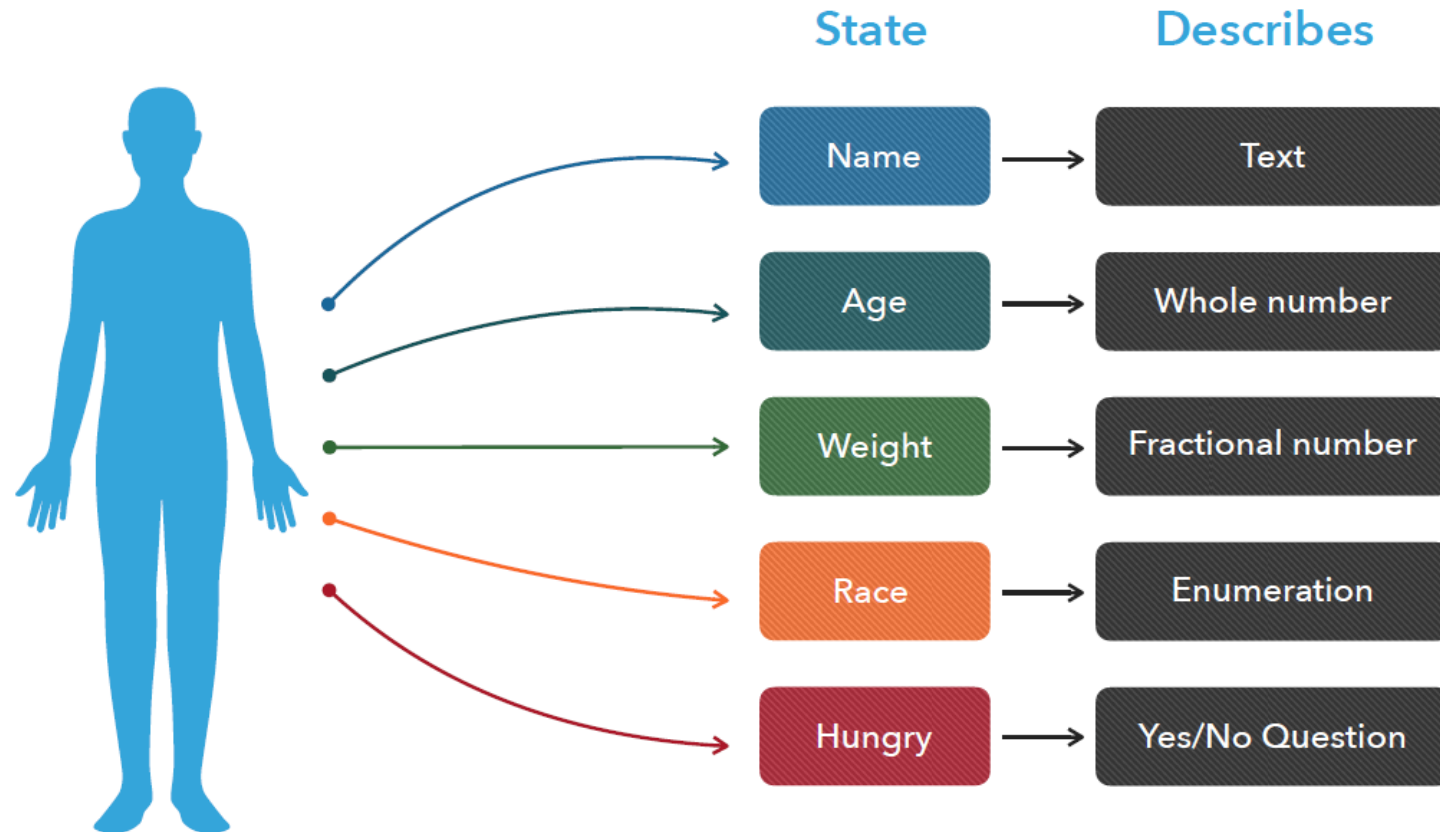
INTRO TO VARIABLES

VARIABLE DEFINITION

- Variable is a named **placeholder** that:
 - Stores data
 - Describes what **type of data** you can store
 - Describes **size or amount** of data it can store



VARIABLE RELATIONSHIP WITH OBJECTS



DATA TYPES

DATA TYPE CATEGORIES

- **Primitive** values
 - **Integer:** byte, short, int, long (e.g. 3, 7, 42, 2018)
 - **Fractional:** float, double (e.g. 3.1415, 2.7, 19.0)
 - **Logical:** boolean (*true or false*)
 - **Textual:** char (e.g. a, b, c, x, y, z)
- **Reference** values
 - Everything else



INTEGER DATA TYPES

| Name | Assignable Values | Size |
|-------|----------------------------|---------|
| byte | -128 ... 127 | 1 byte |
| short | -32,768 ... 32,767 | 2 bytes |
| int | $-2^{31} \dots 2^{31} - 1$ | 4 bytes |
| long | $-2^{63} \dots 2^{63} - 1$ | 8 bytes |



FLOATING POINT DATA TYPES

| Name | Precision | Size |
|--------|-----------|---------|
| float | Single | 4 bytes |
| double | Double | 8 bytes |



LOGICAL DATA TYPE

| Name | Assignable Values | Size |
|---------|-------------------|--------|
| boolean | true / false | 1 byte |



TEXTUAL DATA TYPE

| Name | Assignable Values | Size |
|----------------|--------------------------------------|---------|
| char (unicode) | 0 ('\u0000') ... 65535 ('\uffff') | 2 bytes |



VARIABLE DECLARATION



VARIABLE DECLARATION

- Variable declaration **without** value assignment
- Variable declaration **with** value assignment

```
type name;
```

```
type name = value;
```



VARIABLE DECLARATION

- Variable declaration **without** value assignment
- Variable declaration **with** value assignment

```
int age;
```

```
int age = 25;
```



VARIABLE DECLARATION BREAKDOWN

Variable data type



Assignment operator



`int age = 25;` ← End of statement



Variable value



Variable name



EXAMPLES

```
byte numberOfWheels = 4;  
short selfEsteem = 100;  
int studentsGraduated = 1001;  
long height = 80;  
float pie = 3.14f;  
double weight = 70.5;  
boolean hungry = true;  
char lastLetterOfTheAlphabet = 'Z';
```



VARIABLE NAMING



NAMING RULES

- Any variable is **allowed** to **start** with
 - Letters (**A-Z**)
 - Special characters (**'\$'** - **dollar**, **'_'** - **underscore**)
- Any variable name is **allowed** to **contain**
 - Alphanumeric characters (**A-Z**, **0-9**)
 - Special characters (**'\$'** - **dollar**, **'_'** - **underscore**)
- Variable name is **case-sensitive**
- Java language [keywords](#) or **reserved** words **cannot be used** as variable name



NAMING: DOs

- Single-worded name should be lowercase
- Multi-worded name should
 - **First** word **lowercase**
 - **Subsequent** words start with **capital letters**
 - **No** intervening spaces or punctuation
- Explains the **purpose** of variable



NAMING: DON'Ts

- **Starting** variable name with \$ or _ is highly **discouraged**
- **Avoid** using \$ anywhere in the variable name
- **Avoid appending type information** to the variable name



NAMING DOs AND DON'Ts EXAMPLES

Please, **do**

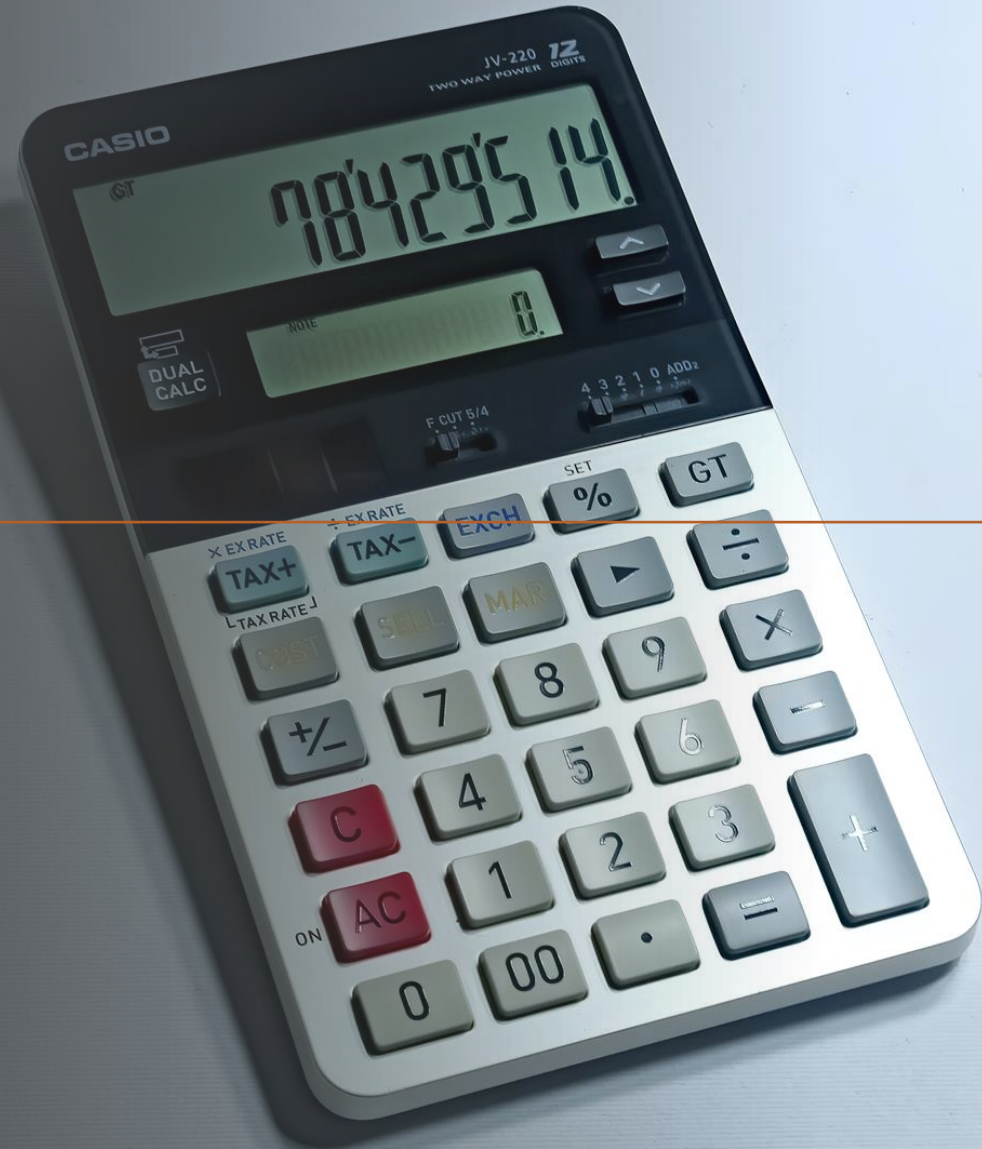
- size, xCoordinate, eyeColor, currentDayOfTheWeek

Please, **don't**

- _counter, \$bankBalance, Timestamp,
- 7daysOfTheWeek, !variableName, *notPointer



ARITHMETIC OPERATORS



ARITHMETIC OPERATORS

| Operator | Operation |
|----------|----------------|
| + | Addition |
| - | Subtraction |
| / | Division |
| * | Multiplication |
| % | Remainder |



ADDITION

Integer Numbers

```
int a = 10;  
int b = 30;  
int result = a + b;
```

```
result == 40
```

Fractional numbers

```
double x = 1.5;  
double y = 2.7;  
double result = x + y;
```

```
result == 4.2
```



SUBTRACTION

Integer Numbers

```
int a = 30;  
int b = 20;  
int result = a - b;
```

```
result == 10
```

Fractional numbers

```
double x = 5.4;  
double y = 1.6;  
double result = x - y;
```

```
result == 3.8
```



MULTIPLICATION

Integer Numbers

```
int a = 2;  
int b = 4;  
int result = a * b;
```

```
result == 8
```

Fractional numbers

```
double x = 2.5;  
double y = 6.4;  
double result = x * y;
```

```
result == 16.0
```



DIVISION

Integer Numbers

```
int a = 10;  
int b = 5;  
int result = a / b;
```

```
result == 2
```

Fractional numbers

```
double x = 18.0;  
double y = 4.8;  
double result = x / y;
```

```
result == 3.75
```



REMINDER

Integer Numbers

```
int a = 9;  
int b = 6;  
int result = a % b;
```

```
result == 3
```

Fractional numbers

```
double x = 10.0;  
double y = 4.5;  
double result = x % y;
```

```
result == 1.00
```



TRICKY QUESTION

Is result integer?

```
int a = 10;  
int b = 4;  
int result = a / b;
```

result == ?

Is result double?

```
int x = 10;  
int y = 4;  
double result = x / y;
```

result == ?



TYPE CASTING

- Operations with widening result require explicit type conversion (casting)

```
int x = 10;  
int y = 4;  
double result = x / (double) y;  
  
result == ?
```



OUTPUT TO CONSOLE



WRITING OUTPUT TO CONSOLE

- Write to the console **content** of the "args" variable
- Write to the console **directly** without using variable

```
System.out.println(args);
```

```
System.out.println("Hello World!");
```

CONSOLE OUTPUT EXAMPLE

Printing variable

```
int a = 10;  
String b = "Hi!";  
System.out.println(a);  
System.out.println(b);
```

10

Hi!

Printing directly

```
System.out.println(374);  
System.out.println("A");
```

374

A



CONSOLE OUTPUT STATEMENT BREAKDOWN

Class with standard
output, input and error
output streams

Function that writes
argument to specified
stream

```
System.out.println("Hello World!");
```

Inner class that helps
to specify stream flow
direction

Argument to be written
to the stream



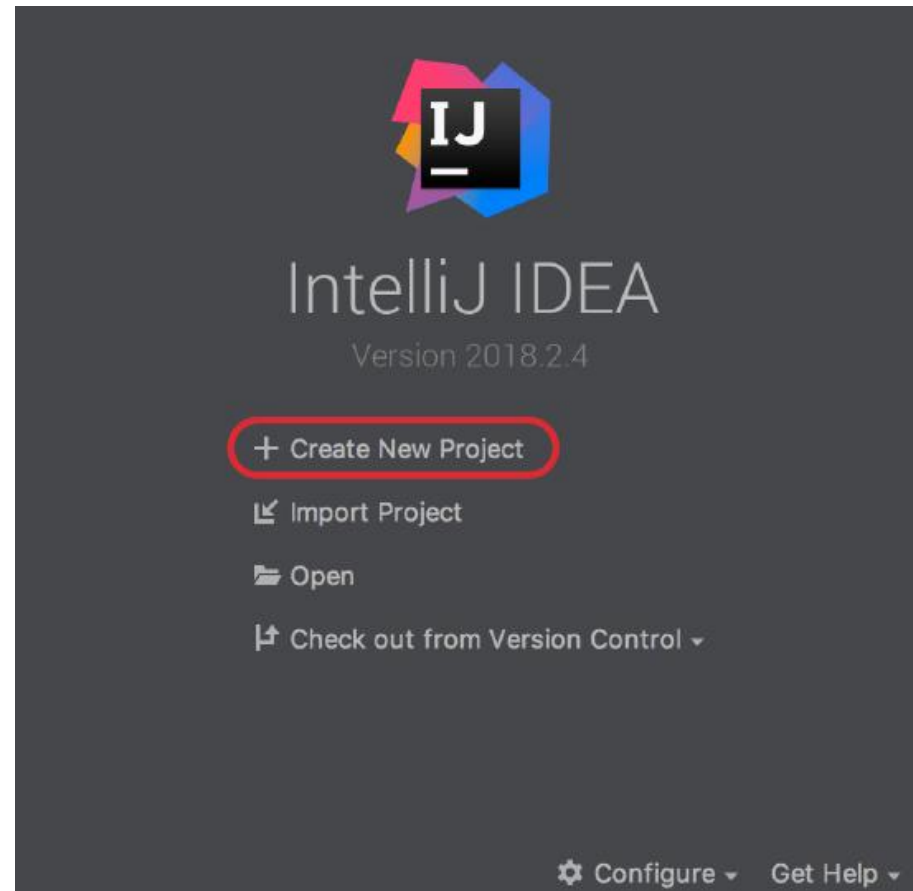
**CREATE FIRST
PROJECT**

IJ
—

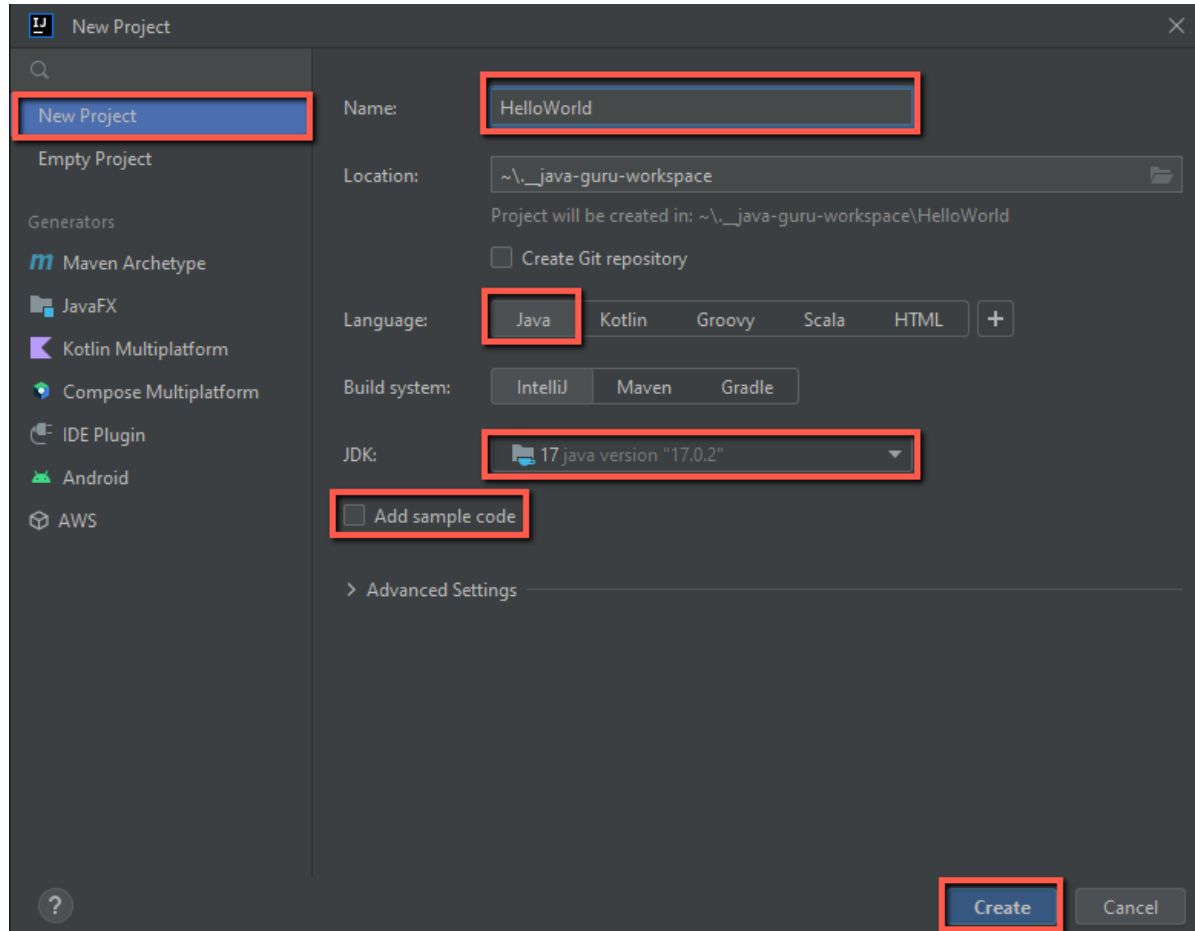
RUN INTELLIJ IDEA



WELCOME SCREEN



PROJECT CONFIGURATION

A screenshot of the IntelliJ IDEA "New Project" dialog box. The dialog is dark-themed. On the left, a sidebar lists options: "New Project" (highlighted with a red box), "Empty Project", and a "Generators" section with icons for "Maven Archetype", "JavaFX", "Kotlin Multiplatform", "Compose Multiplatform", "IDE Plugin", "Android", and "AWS". The main area contains configuration fields: "Name:" with the text "HelloWorld" (boxed in red); "Location:" with a text field showing "~\._java-guru-workspace" and a folder icon; a line of text "Project will be created in: ~\._java-guru-workspace\HelloWorld"; an unchecked checkbox "Create Git repository"; "Language:" with buttons for "Java" (boxed in red), "Kotlin", "Groovy", "Scala", "HTML", and a "+" button; "Build system:" with buttons for "IntelliJ" (selected), "Maven", and "Gradle"; "JDK:" with a dropdown menu showing "17 java version '17.0.2'" (boxed in red); and an unchecked checkbox "Add sample code" (boxed in red). At the bottom, there is a "> Advanced Settings" link and two buttons: "Create" (boxed in red) and "Cancel".

New Project

New Project

Empty Project

Generators

- Maven Archetype
- JavaFX
- Kotlin Multiplatform
- Compose Multiplatform
- IDE Plugin
- Android
- AWS

Name: HelloWorld

Location: ~\._java-guru-workspace

Project will be created in: ~\._java-guru-workspace\HelloWorld

☐ Create Git repository

Language: Java Kotlin Groovy Scala HTML +

Build system: IntelliJ Maven Gradle

JDK: 17 java version "17.0.2"

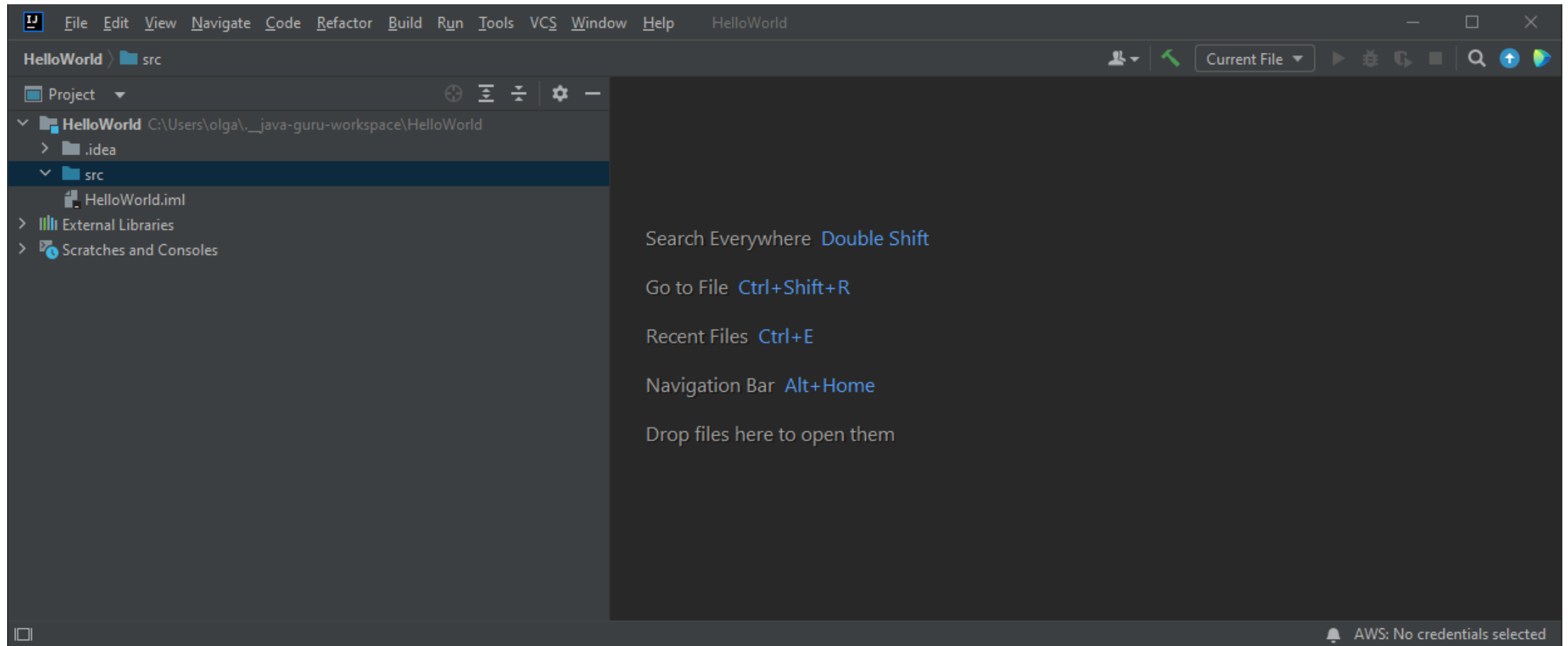
☐ Add sample code

> Advanced Settings

Create Cancel



PROJECT OVERVIEW



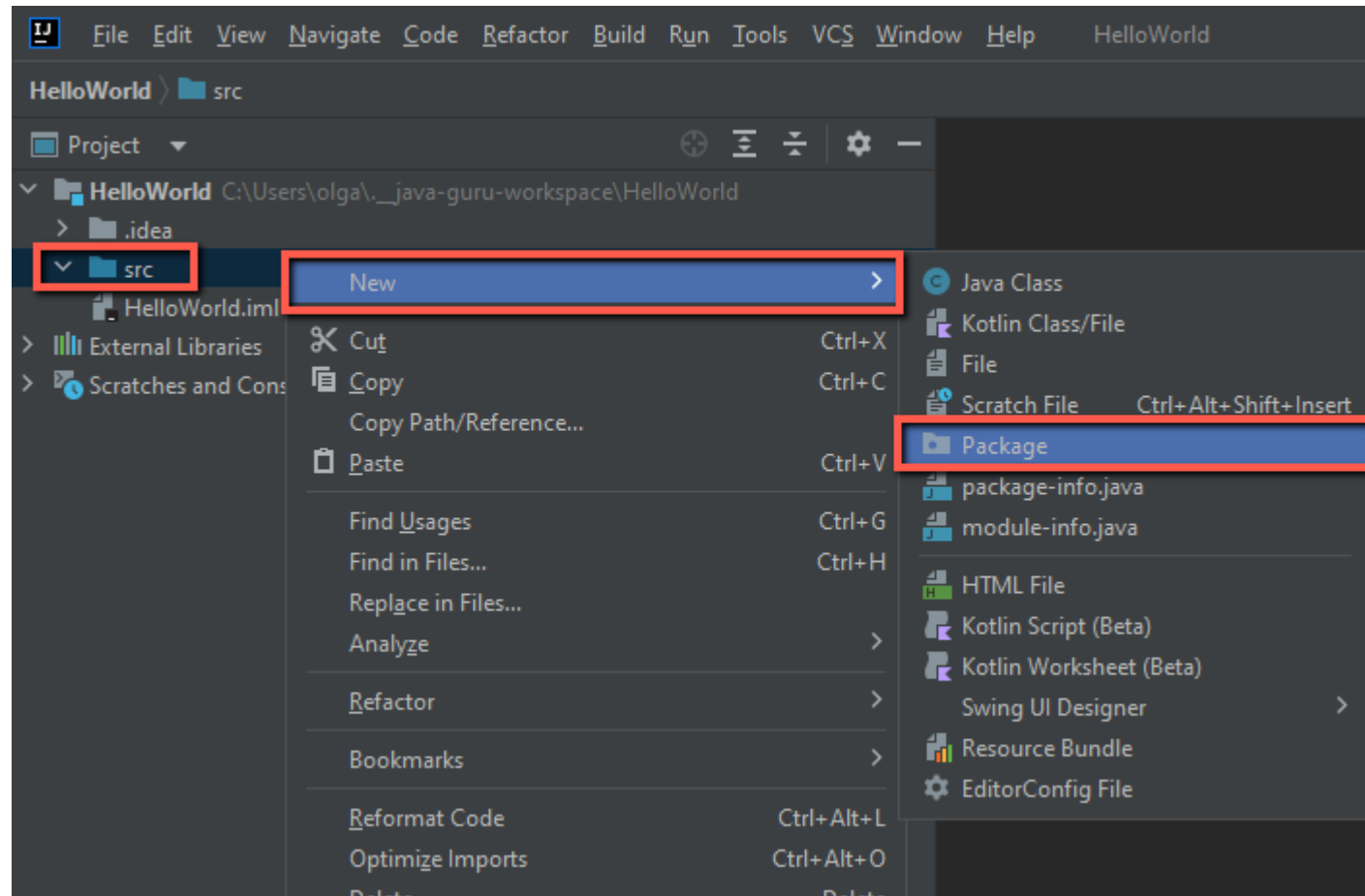
FIRST JAVA APPLICATION

OBJECTIVE

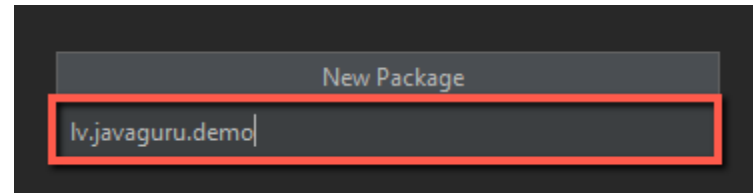
- **Print** «Hello, World!» text to the console



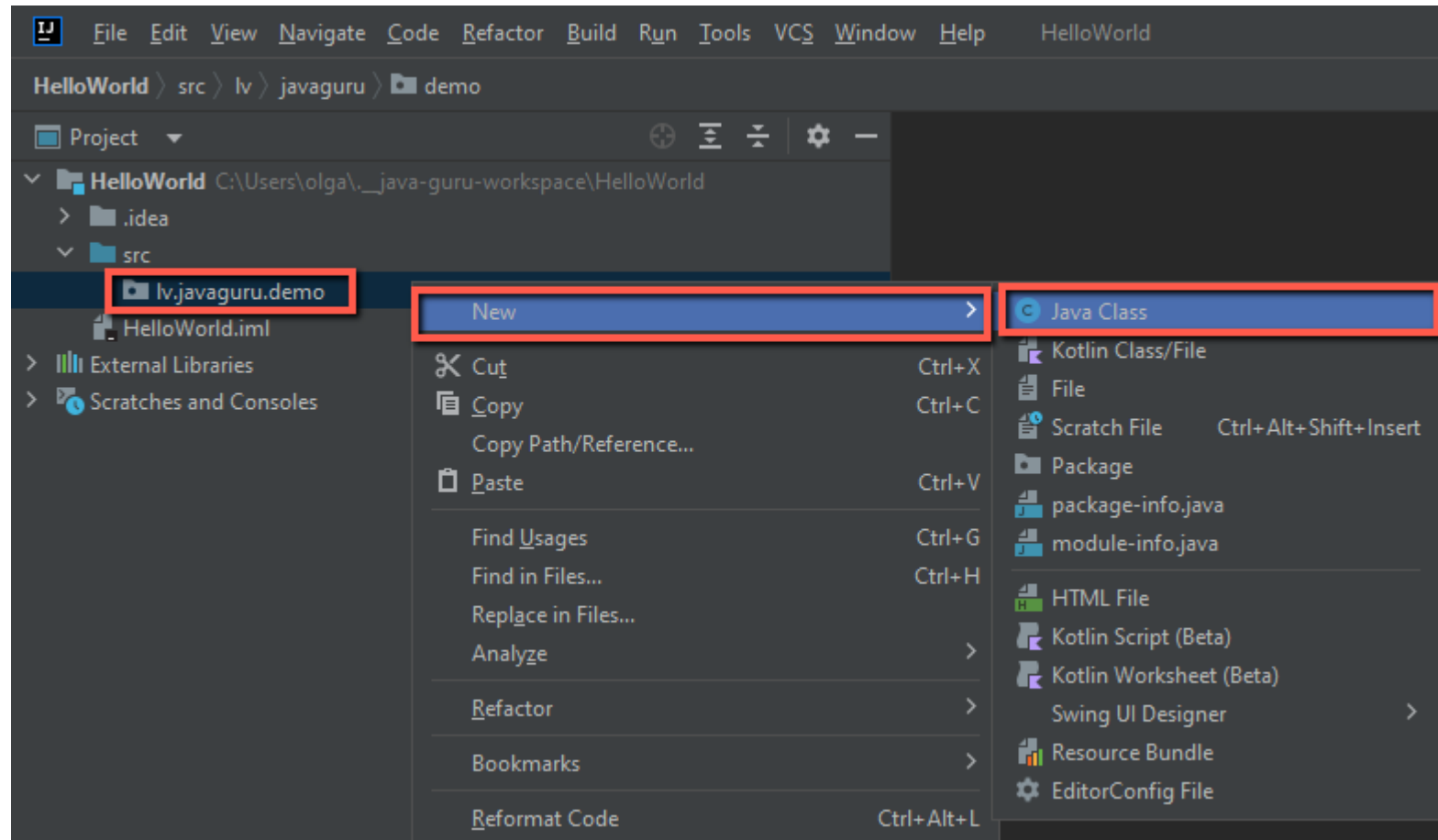
CREATE PACKAGE



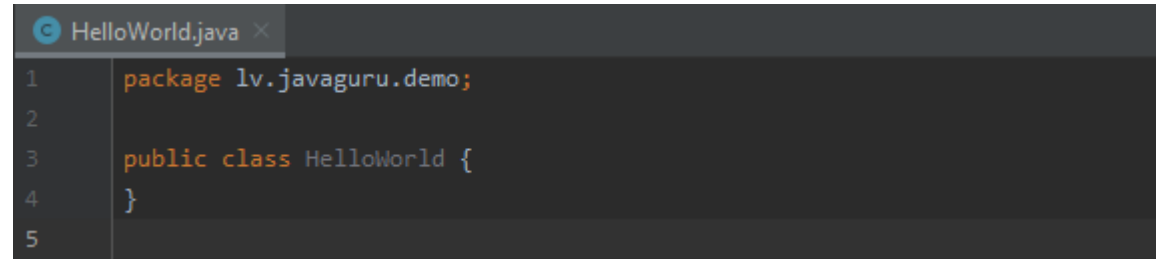
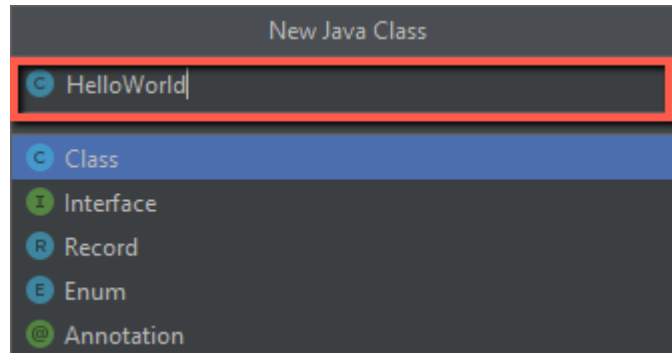
PACKAGE NAME



CREATE CLASS



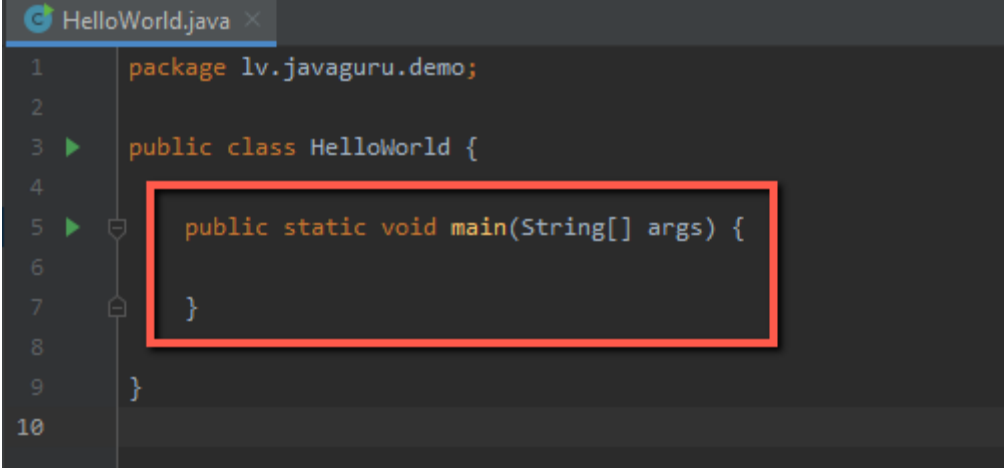
CLASS NAME



```
1 package lv.javaguru.demo;  
2  
3 public class HelloWorld {  
4 }  
5
```



APPLICATION ENTRY



```
1 package lv.javaguru.demo;  
2  
3 public class HelloWorld {  
4  
5     public static void main(String[] args) {  
6  
7     }  
8  
9 }  
10
```

The screenshot shows a code editor with a file named `HelloWorld.java`. The code defines a package `lv.javaguru.demo`, a public class `HelloWorld`, and a public static method `main` that takes a `String[] args` parameter. The `main` method body is currently empty. A red rectangular box highlights the `main` method signature and its opening curly brace, indicating it as the application's entry point.

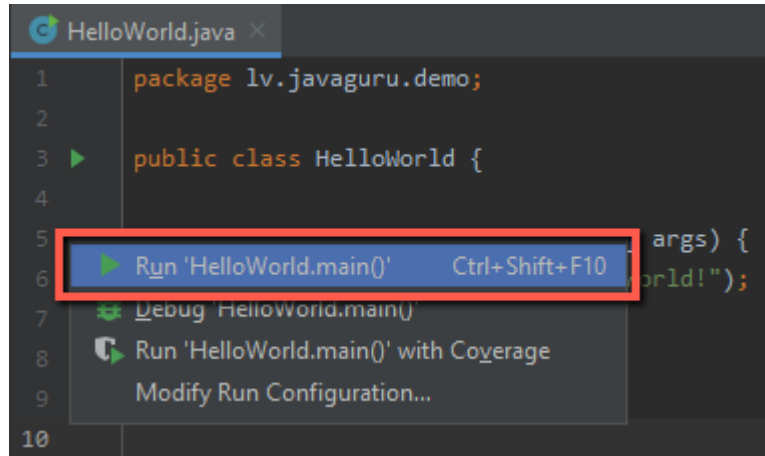


PRINT «HELLO, WORLD!»

```
HelloWorld.java x
1  package lv.javaguru.demo;
2
3  ▶ public class HelloWorld {
4
5  ▶   public static void main(String[] args) {
6      System.out.println("Hello, World!");
7   }
8
9  }
10
```



RUN APPLICATION

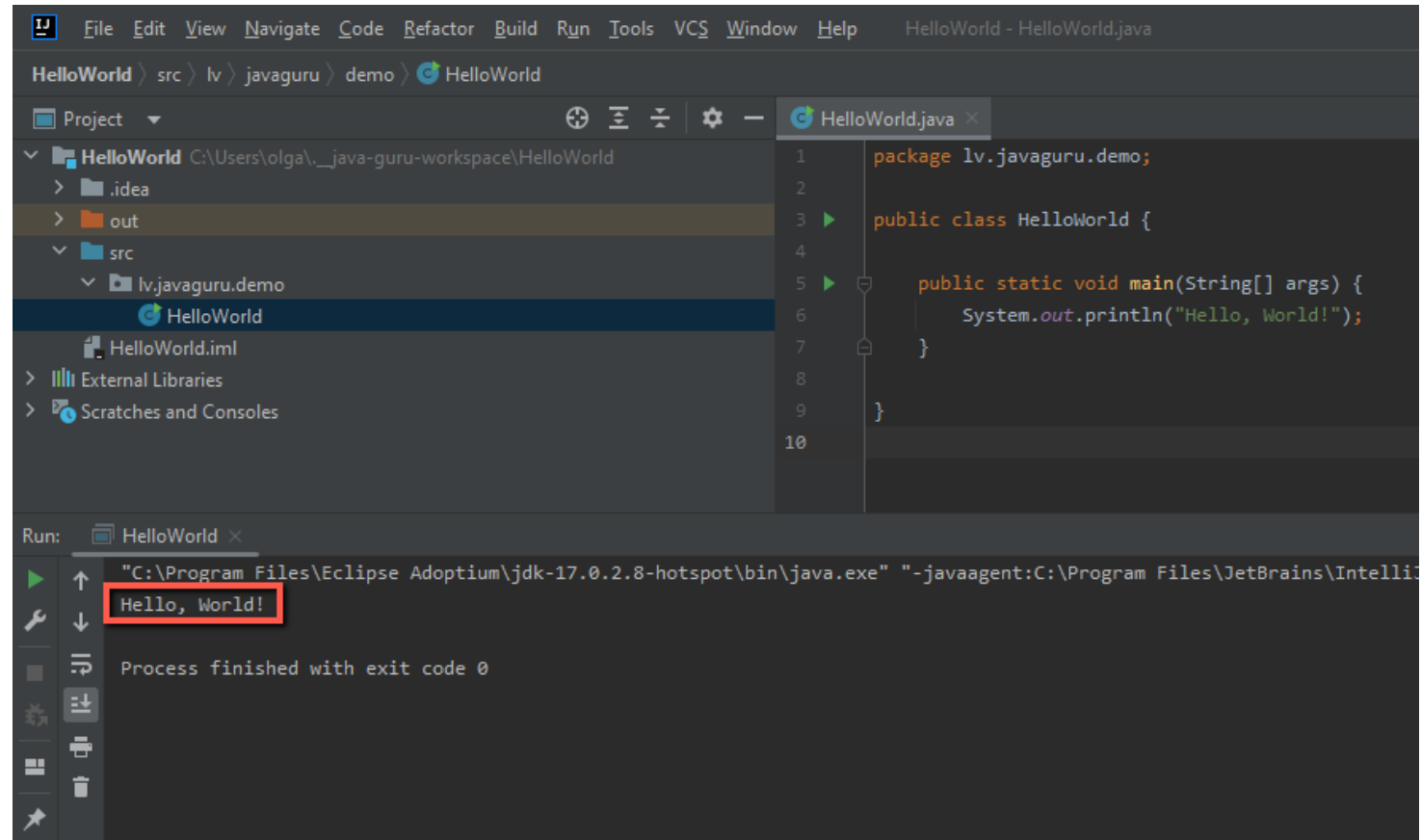


```
1 package lv.javaguru.demo;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         System.out.println('Hello World!');
7     }
8 }
9
10
```

- Run 'HelloWorld.main()' Ctrl+Shift+F10
- Debug 'HelloWorld.main()'
- Run 'HelloWorld.main()' with Coverage
- Modify Run Configuration...



APPLICATION OUTPUT

A screenshot of an IDE window. The top part shows a project explorer on the left with a tree view containing 'HelloWorld', 'src', 'out', and 'HelloWorld.iml'. The main editor area displays a Java file 'HelloWorld.java' with the following code:

```
1 package lv.javaguru.demo;
2
3 public class HelloWorld {
4
5     public static void main(String[] args) {
6         System.out.println("Hello, World!");
7     }
8
9 }
10
```

The bottom part of the window shows a 'Run' console. It contains the command: `"C:\Program Files\Eclipse Adoptium\jdk-17.0.2.8-hotspot\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ..."` and the output: `Hello, World!`, which is highlighted with a red rectangle. Below the output, it says 'Process finished with exit code 0'.

TASK 1: OBJECTIVES

1. The program should define two integer variables
2. Calculate the sum of these variables
3. Print result to the console



TASK 1: CODE

1. Create 1st variable and assign it value of 10
2. Create 2nd variable and assign it value of 20
3. Assign 3rd variable result of computed sum
4. Print 3rd variable to the console

```
package lv.javaguru.demo;  
  
public class Calculator {  
  
    public static void main(String[] args) {  
        ① int firstNumber = 10;  
        ② int secondNumber = 20;  
        ③ int sumResult = firstNumber + secondNumber;  
        ④ System.out.println(sumResult);  
    }  
}
```

TASK 2: OBJECTIVES

1. The program should create random number generator
2. Generate random number within 0 - 100 range inclusive
3. Print result to the console



TASK 2: CODE

1. Create random generator and assign it to the variable
2. Generate randomNumber and assign result to the variable
3. Print randomNumber variable to the console

```
package lv.javaguru.demo;

import java.util.Random;

public class RandomNumberGenerator {

    public static void main(String[] args) {
        ① Random randomGenerator = new Random();
        ② int randomNumber = randomGenerator.nextInt();
        ③ System.out.println(randomNumber);
    }

}
```


REFERENCES

REFERENCES

- <https://docs.oracle.com/javase/tutorial/java/nutsandbolts/datatypes.html>
- <http://tutorials.jenkov.com/java/variables.html>
- <http://tutorials.jenkov.com/java/data-types.html>
- <https://javapapers.com/core-java/system-out-println/>



QUESTIONS?



THANK YOU!

