

Java Basics Tutorial

Part 2: IntelliJ IDEA



Free Code Lessons



IDE

Integrated Development Environments



- To write programs, we need an **IDE**
 - For **Java** →
 - **IntelliJ IDEA**
 - **Eclipse**
 - **NetBeans**
 - For **C#** → **Visual Studio** / **Raider** / **VS Code**
 - For **Python** → **PyCharm** / **Eclipse** / **VS Code**
 - For **JavaScript** → **WebStorm** / **VS Code**



- **Integrated Development Environments (IDE)** simplify software development
- **IDEs** save time and eases the process of coding
 - Writing, running, testing, and debugging the code
- In our **Java** lessons, we shall use **IntelliJ IDEA Community**
 - Free, open-source IDE from JetBrains
 - <https://jetbrains.com/idea>





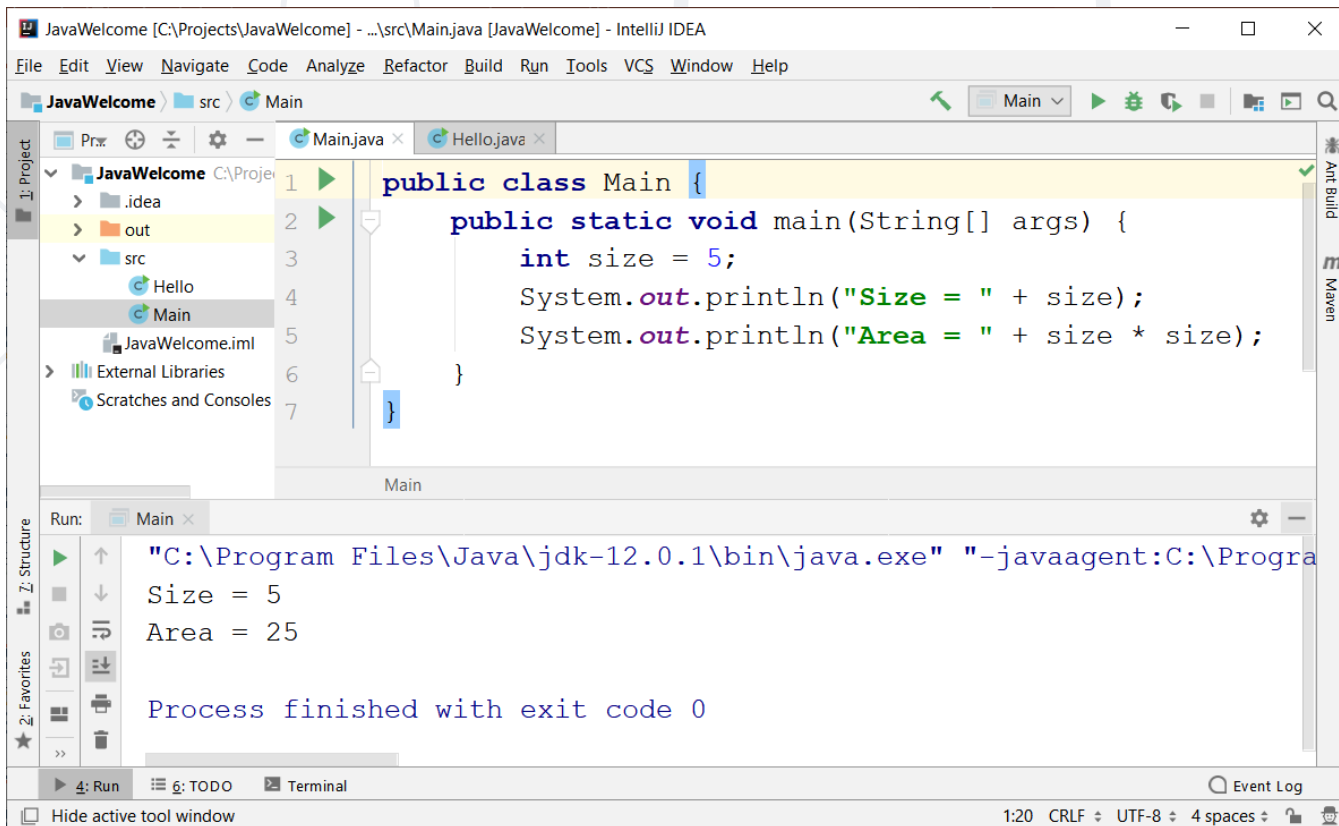
```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int dollars = scanner.nextInt();  
        double euro = dollars * 0.853758697;  
        System.out.println("Euro: " + euro);  
    }  
}
```

IntelliJ IDEA

**Powerful IDE for
Java Developers**

What is IntelliJ IDEA ?

- **IntelliJ IDEA** is a powerful IDE for Java developers
 - Runs in Windows, Linux and macOS



The screenshot displays the IntelliJ IDEA interface. The main editor shows a Java class named `Main` with the following code:

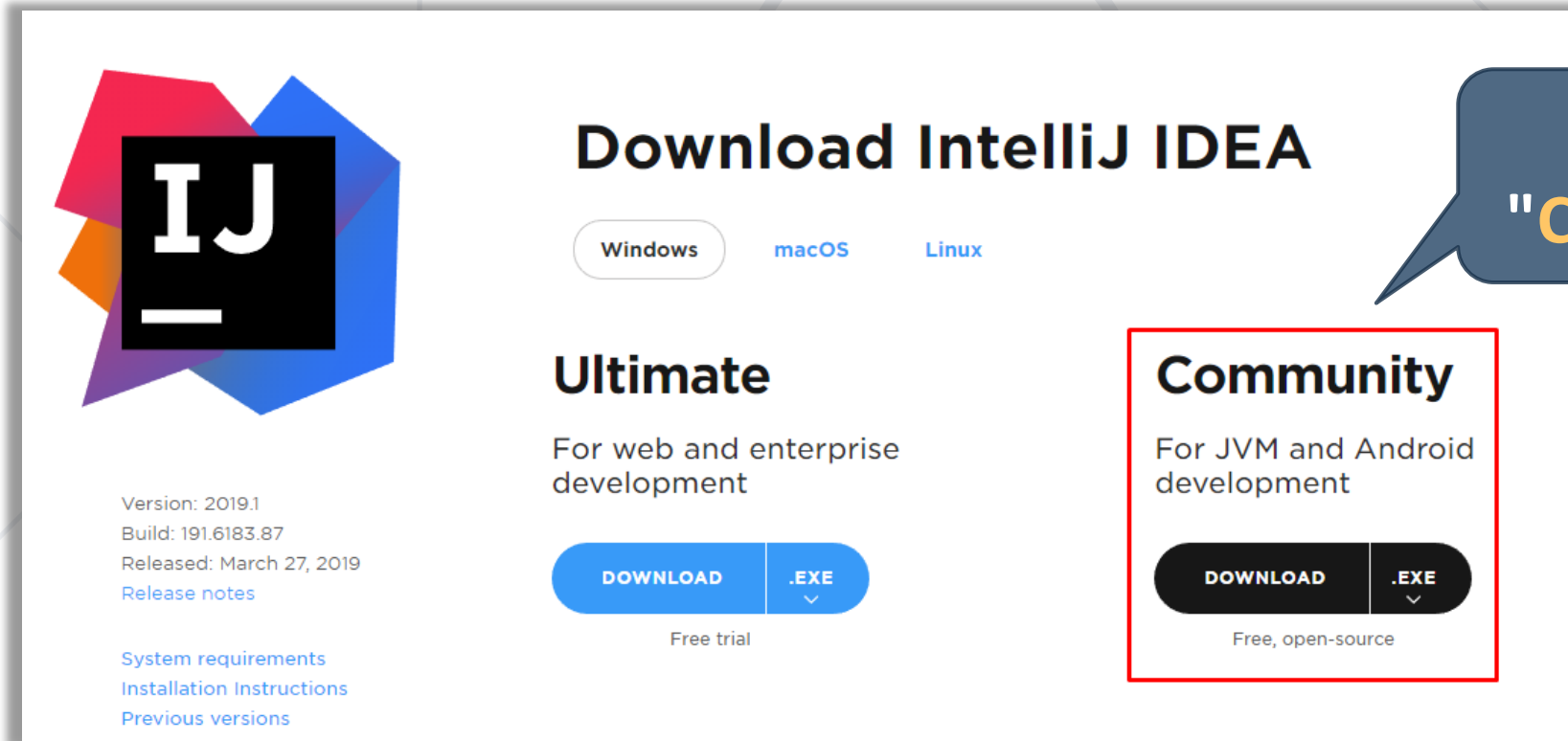
```
public class Main {  
    public static void main(String[] args) {  
        int size = 5;  
        System.out.println("Size = " + size);  
        System.out.println("Area = " + size * size);  
    }  
}
```

The Run window at the bottom shows the execution output:

```
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:C:\Progra  
Size = 5  
Area = 25  
  
Process finished with exit code 0
```



- Download the IntelliJ IDEA installer and follow the instructions



The screenshot shows the IntelliJ IDEA download page. On the left is the IntelliJ logo. The main heading is "Download IntelliJ IDEA". Below it are tabs for "Windows", "macOS", and "Linux". There are two main options: "Ultimate" (for web and enterprise development) and "Community" (for JVM and Android development). The "Community" option is highlighted with a red border. A blue callout bubble points to the "Community" option with the text "Choose the free 'Community' edition".

Download IntelliJ IDEA

Windows macOS Linux

Ultimate
For web and enterprise development
Free trial

Community
For JVM and Android development
Free, open-source

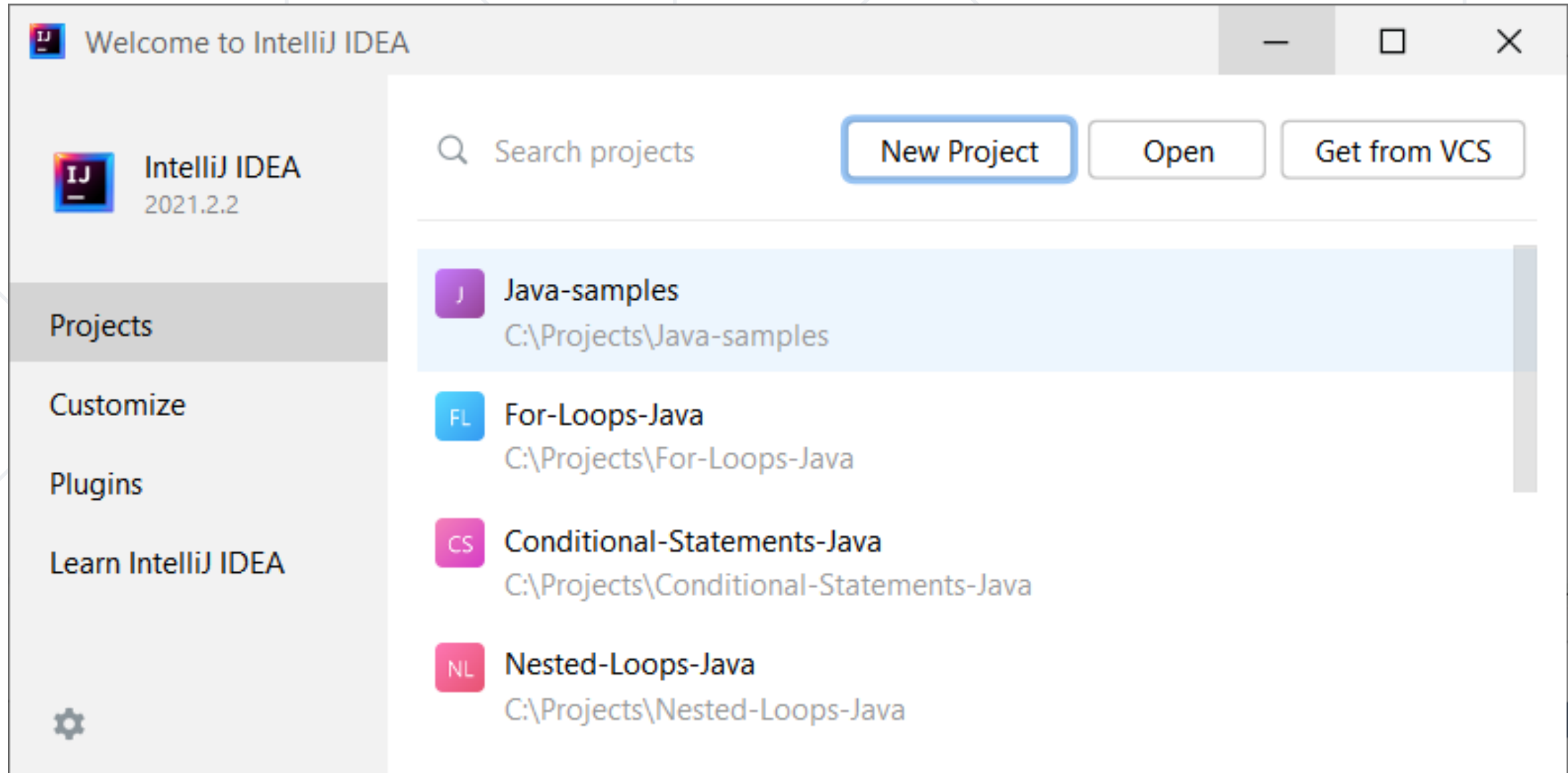
Version: 2019.1
Build: 191.6183.87
Released: March 27, 2019
Release notes

System requirements
Installation Instructions
Previous versions

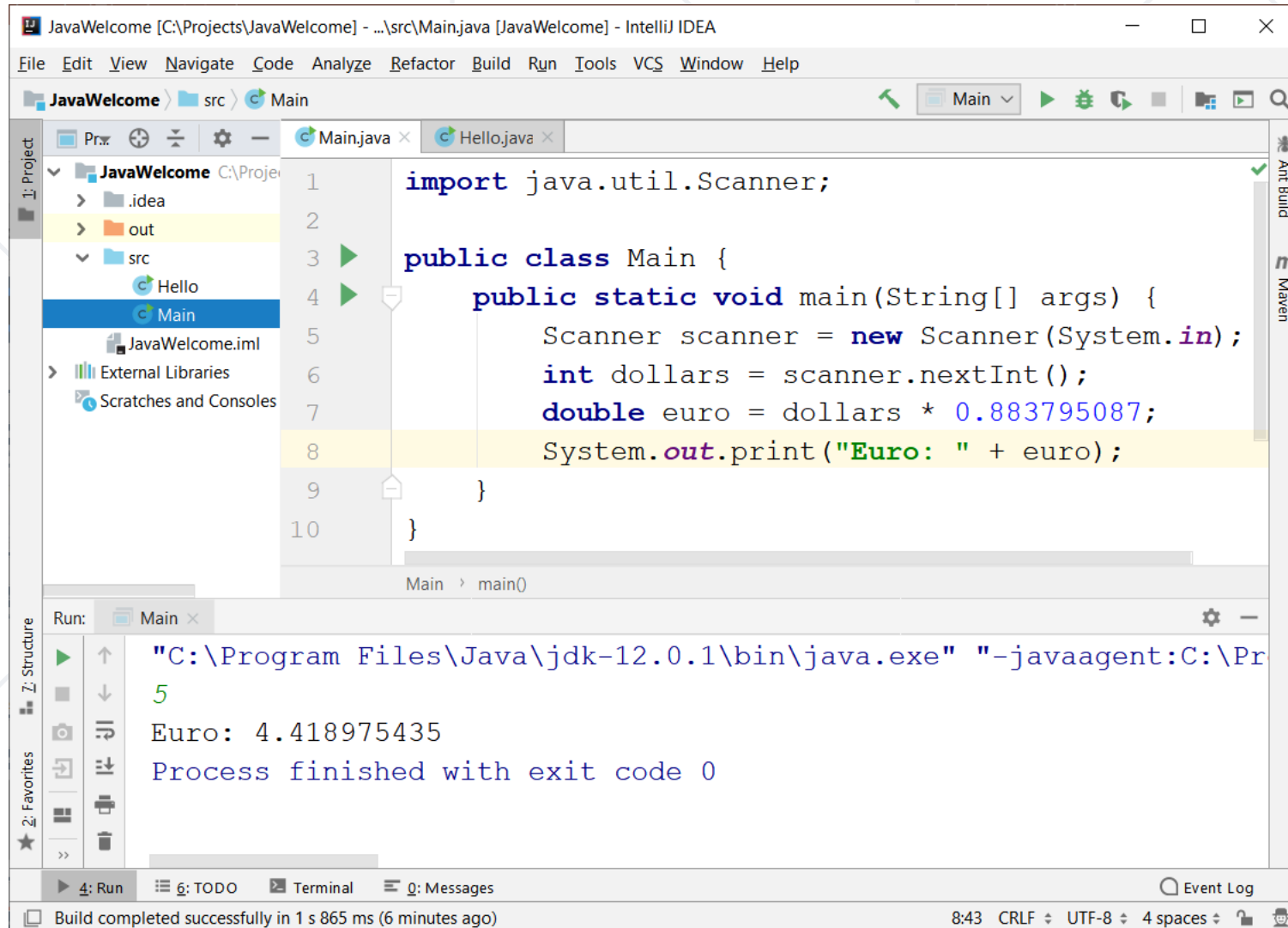
Choose the free "Community" edition

<https://www.jetbrains.com/idea/download>

Running IntelliJ IDEA



Projects in IntelliJ IDEA



- The project explorer holds the **project files**
- The project source code consists of **.java** files (Java classes)

Creating a Console App

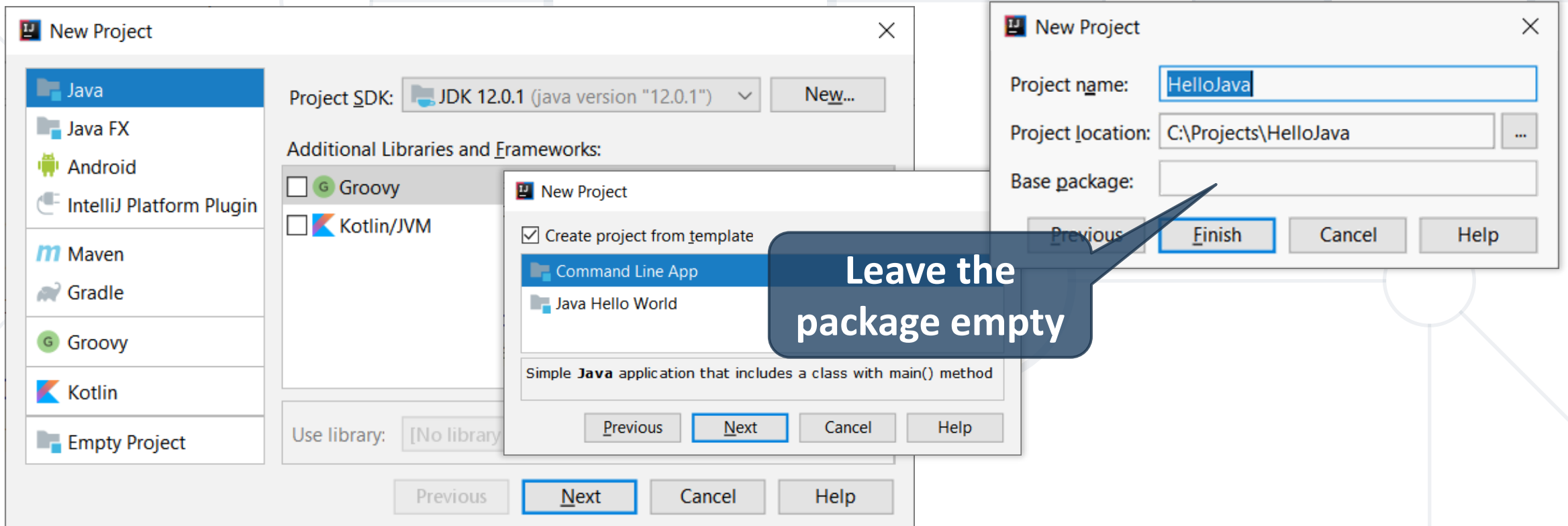
**Create and Run
a Java Console App
in IntelliJ IDEA**



Creating a Console Application in IntelliJ IDEA

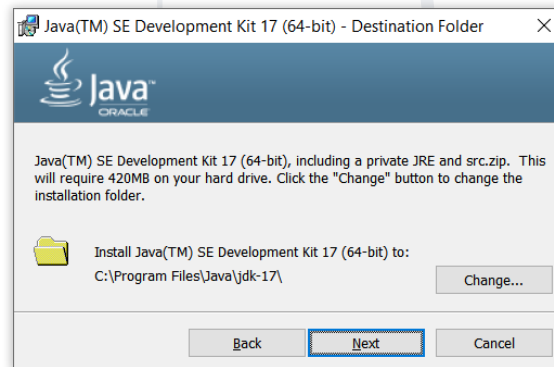
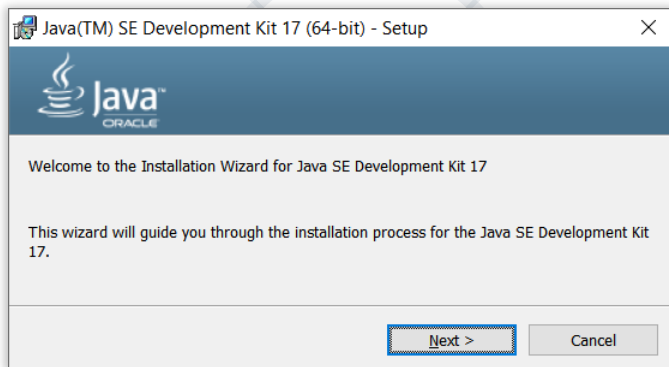


- Create new Console Application in IntelliJ IDEA: **[New Project]** → **[Java]** → **[Template Command Line App]** → **[Finish]**



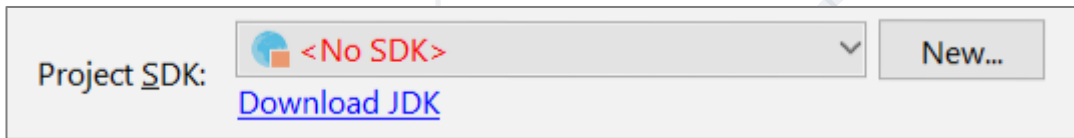
JDK (Java Development Kit)

- **JDK (Java Development Kit) is the Java platform for developers**
 - Java runtime (JVM + libraries) + Java compiler + tools
- **Download and install the latest JDK**
 - <https://oracle.com/java/technologies/downloads> (Oracle)
 - <https://adoptium.net> (Eclipse Adoptium)

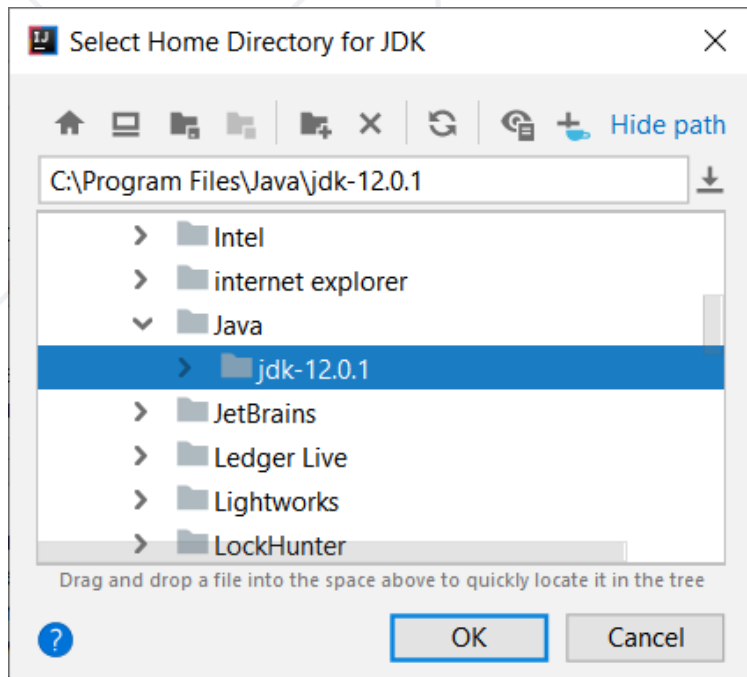


Configuring JDK in IntelliJ IDEA

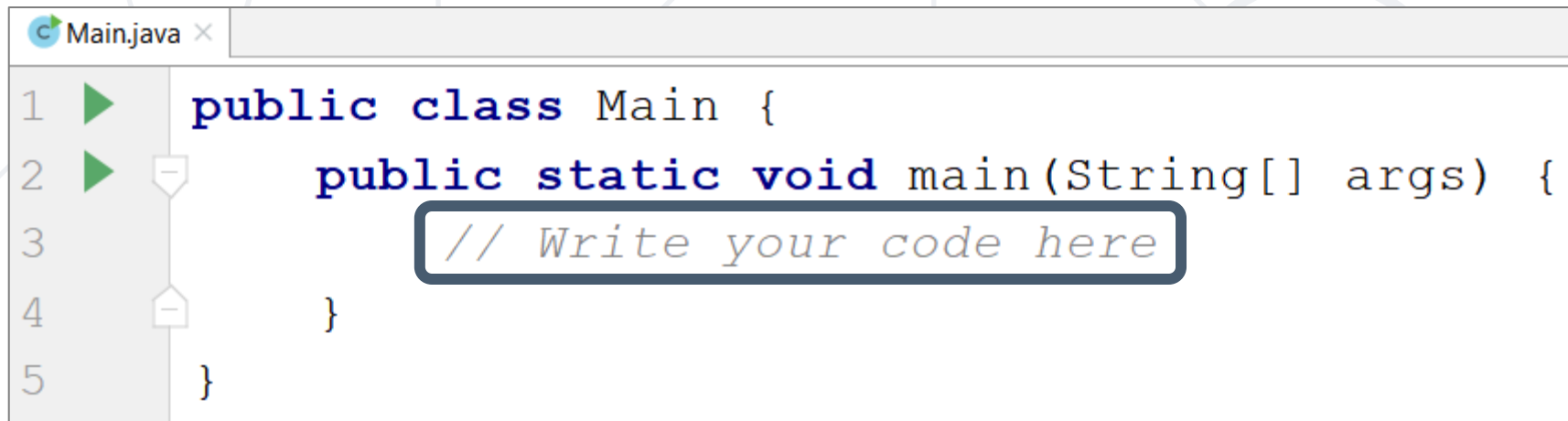
- If no JDK is still configured, you should configure it:



- Click **[New]** button and locate your JDK installation:



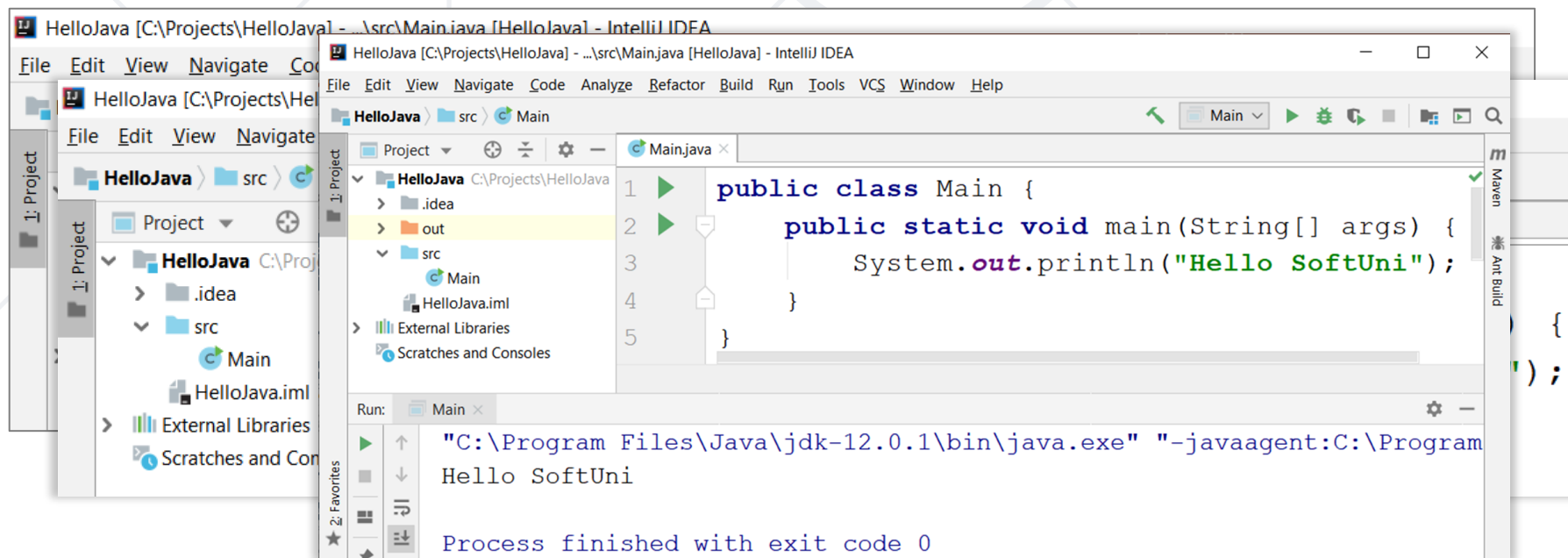
- In Java programs, commands are written in `main(String[] args)`
 - Between the opening and closing curly brackets `{ }`
- Press [**Enter**] after the opening bracket `{` and type your commands
- The code of the program is written with **indentation**



```
1 public class Main {
2     public static void main(String[] args) {
3         // Write your code here
4     }
5 }
```

Starting the Program in IntelliJ IDEA

- To start the program, press **[Ctrl + Shift + F10]**
 - If there are no errors, the program will be launched
- The result will appear in the console (terminal window):



```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("Hello SoftUni");  
    }  
}
```

Run: Main ×

```
"C:\Program Files\Java\jdk-12.0.1\bin\java.exe" "-javaagent:C:\Program  
Hello SoftUni  
Process finished with exit code 0
```


Coding Exercises

**Writing Java
Code with
IntelliJ IDEA**



Learn by Doing

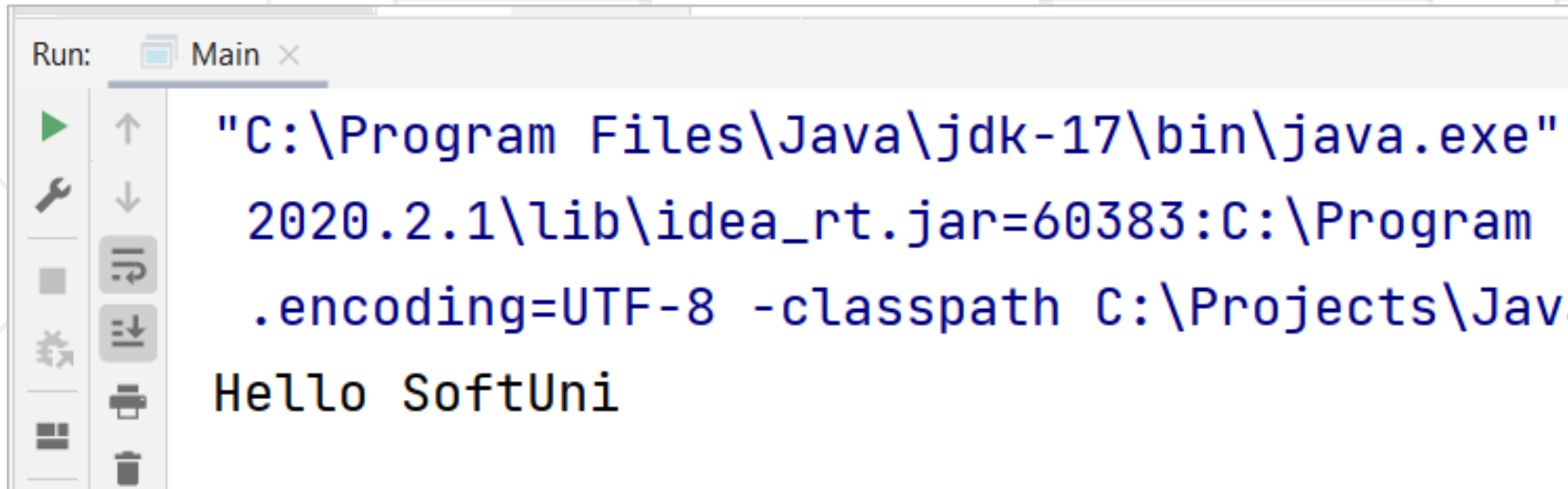
- To learn coding, you need to **write code!**
- Watching **videos** gives you only **knowledge**
- Solving the **exercises**, gives you **experience** and **practical skills**



**Write and submit the
coding exercises!**

Problem: Print "Hello SoftUni"

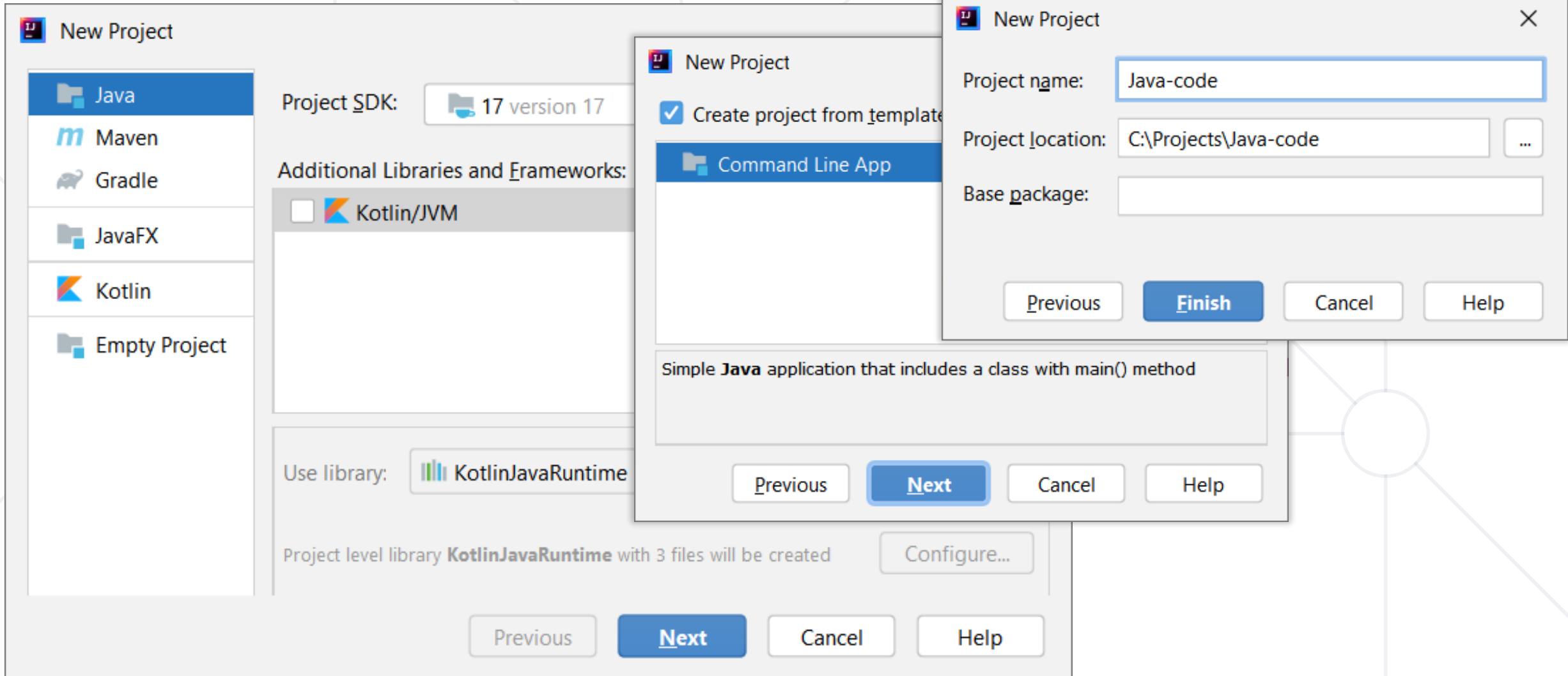
- Write a Java program, which:
 - Prints "Hello SoftUni" on the console



```
Run: Main x
"C:\Program Files\Java\jdk-17\bin\java.exe"
2020.2.1\lib\idea_rt.jar=60383:C:\Program
.encoding=UTF-8 -classpath C:\Projects\Jav
Hello SoftUni
```

- Submit your solution in the SoftUni judge:
<https://judge.softuni.org/Contests/Practice/Index/3252>

Solution: Print "Hello SoftUni"



The image shows the IntelliJ IDEA 'New Project' wizard with three overlapping windows. The background window is the main wizard, the middle window is the 'New Project' dialog for selecting a template, and the foreground window is the 'New Project' dialog for entering project details.

Main 'New Project' Wizard:

- Project SDK: 17 version 17
- Additional Libraries and Frameworks: Kotlin/JVM
- Use library: KotlinJavaRuntime
- Project level library KotlinJavaRuntime with 3 files will be created
- Buttons: Previous, Next, Cancel, Help

'New Project' Dialog (Template Selection):

- Project name: Java-code
- Project location: C:\Projects\Java-code
- Base package: (empty)
- Buttons: Previous, Finish, Cancel, Help

'New Project' Dialog (Description):

- Selected template: Command Line App
- Description: Simple Java application that includes a class with main() method
- Buttons: Previous, Next, Cancel, Help

Solution: Print "Hello SoftUni" (2)

A screenshot of the IntelliJ IDEA IDE interface. The main editor window shows the source code for `HelloSoftUni.java`. The code is as follows:

```
1 public class HelloSoftUni {  
2     public static void main(String[] args) {  
3         System.out.println("Hello SoftUni");  
4     }  
5 }
```

The `main` method is highlighted in yellow. A context menu is open over the code, showing options: `Run 'Main'` (with keyboard shortcut `Ctrl+Shift+F10`), `Debug 'Main'`, `Run 'Main' with Coverage`, and `Modify Run Configuration...`. The `Run 'Main'` option is selected. Below the editor, the Run tool window is open, displaying the command line used to execute the program and the output. The command line is: `"C:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.1\lib\idea_rt.jar=60383:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.1\bin" -Dfile.encoding=UTF-8 -classpath C:\Projects\Java-samples\out\production\Java-samples Main`. The output is: `Hello SoftUni`. The status bar at the bottom indicates "Build completed successfully in 2 sec, 830 ms (15 minutes ago)".

Submission in the Judge System



← → ↻ judge.soft

Hello SoftU

```
1 public class He
2     public stat
3         System.
4     }
5 }
6
```

Submissions

⏪ ⏩ 1 ⏪ ⏩ ↻

Points	Time and memory used	Submission date	
✓ 100 / 100	Memory: 1.13 MB Time: 0.001 s	17:36:50 22.09.2021	Details
Not processed yet	---	17:36:50 22.09.2021	Details

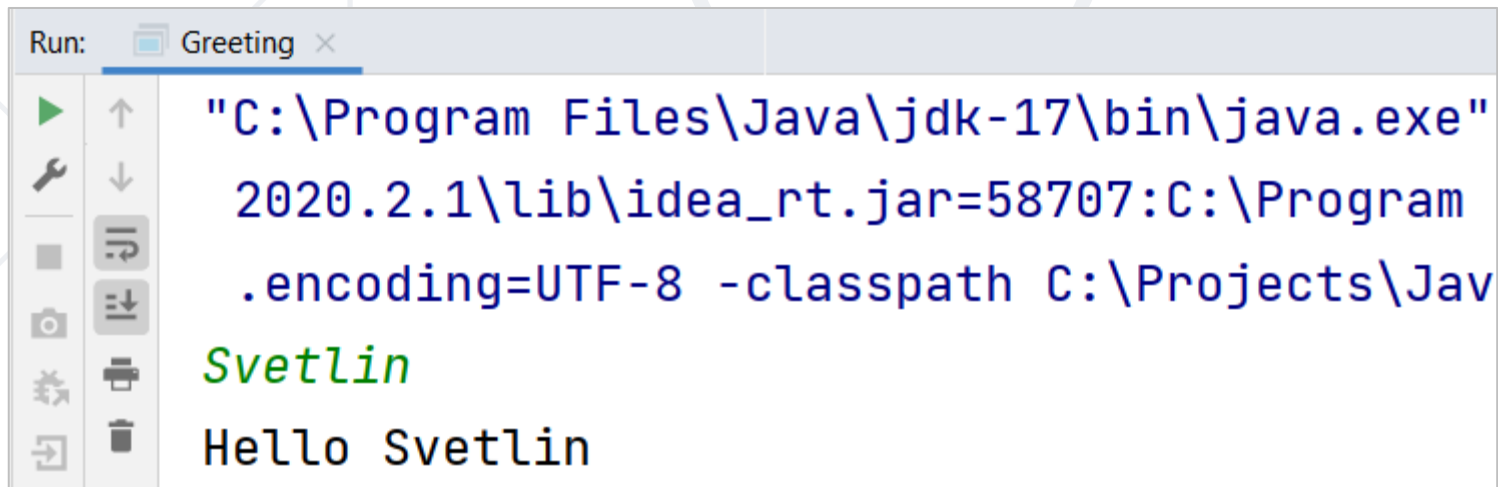
Allowed working time: 0.100 sec.
Allowed memory: 16.00 MB
Size limit: 16.00 KB
Checker: Trim ?

Java code Submit

<https://judge.softuni.org/Contests/Practice/Index/3252>

Problem: Greeting

- Write a Java program, which:
 - Reads from the console your name
 - Prints "Hello <<*name*>>"



```
Run: Greeting x
"C:\Program Files\Java\jdk-17\bin\java.exe"
 2020.2.1\lib\idea_rt.jar=58707:C:\Program
.encoding=UTF-8 -classpath C:\Projects\Jav
Svetlin
Hello Svetlin
```

- Submit your solution in the SoftUni judge:
<https://judge.softuni.org/Contests/Practice/Index/3252>

Creating a New Java Class



The screenshot illustrates the process of creating a new Java class in an IDE. It features two overlapping windows. The top window, titled 'Java-samples - HelloSoftUni.java', shows the source code for a class named 'HelloSoftUni'. The code is as follows:

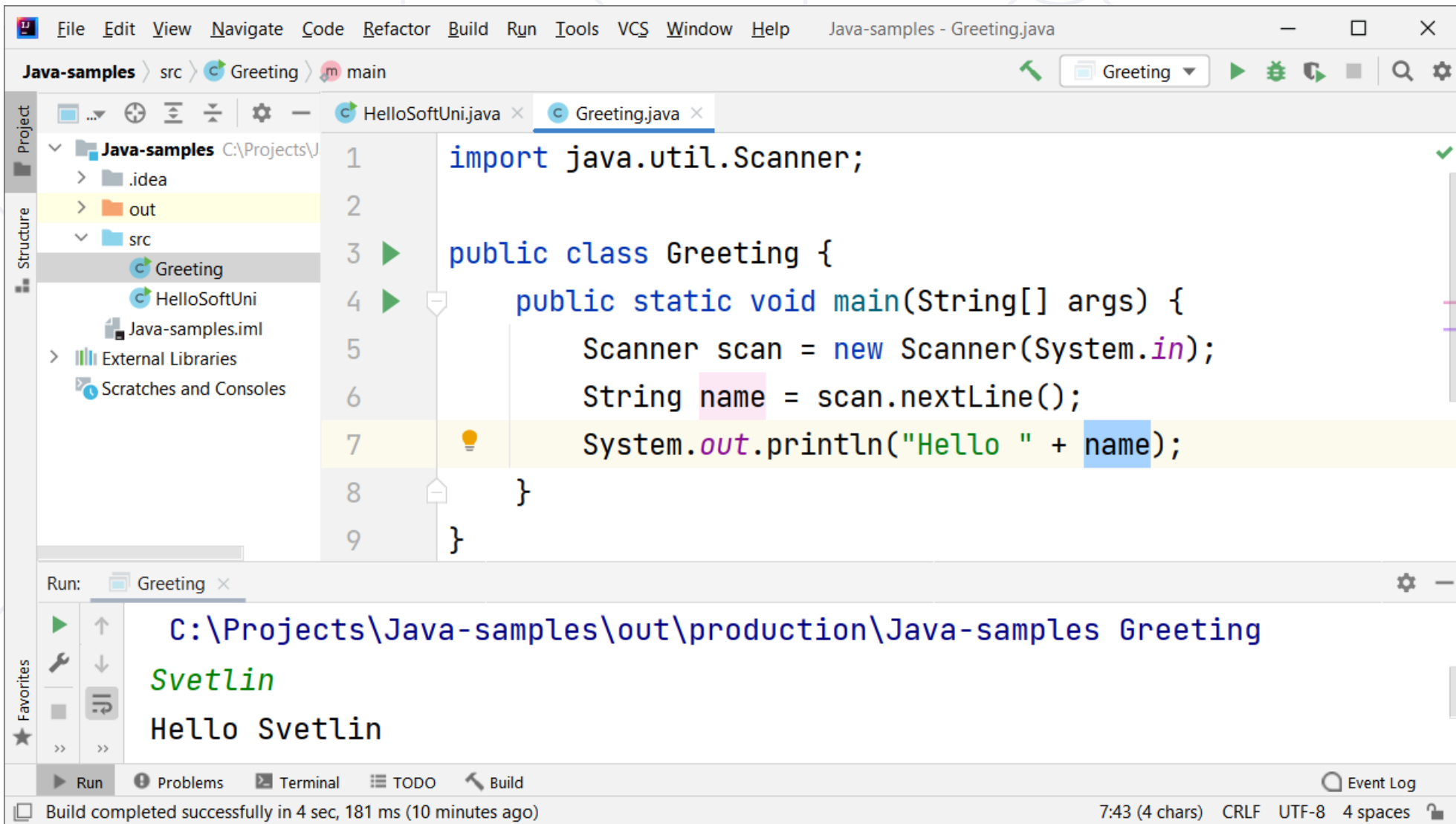
```
1 public class HelloSoftUni {  
2     public static void main(String[] args) {  
3         System.out.println("Hello SoftUni");  
4     }  
5 }
```

The bottom window, titled 'Java-samples - Greeting.java', shows the source code for a new class named 'Greeting'. The code is as follows:

```
1 public class Greeting {  
2     }  
3 }
```

The IDE interface includes a menu bar with options like File, Edit, View, Navigate, Code, Refactor, Build, Run, Tools, VCS, Window, and Help. A 'New' menu is open, showing options for creating a 'Java Class', 'Kotlin Class/File', 'Interface', 'Record', 'Enum', and 'Annotation'. The 'Project' and 'Structure' toolbars are also visible, showing the current project structure and the newly created 'Greeting' class.

Solution: Greeting



The screenshot displays an IDE window titled "Java-samples - Greeting.java". The main editor shows the following Java code:

```
1 import java.util.Scanner;
2
3 public class Greeting {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         String name = scan.nextLine();
7         System.out.println("Hello " + name);
8     }
9 }
```

The code is executed, and the output is shown in the Run window:

```
C:\Projects\Java-samples\out\production\Java-samples Greeting
Svetlin
Hello Svetlin
```

The status bar at the bottom indicates: "Build completed successfully in 4 sec, 181 ms (10 minutes ago)" and "7:43 (4 chars) CRLF UTF-8 4 spaces".

Problem: Customer Info

- Write a program, which **reads** customer's **first name**, **last name** and **email** and prints the read data in the following format:
 - Customer: <<*fname*>> <<*lname*>> (<<*email*>>)



```
Run: CustomerInfo x
Jessica
Green
jessy@gmail.com
Customer: Jessica Green (jessy@gmail.com)
```

- Submit your solution in the SoftUni judge:
<https://judge.softuni.org/Contests/Practice/Index/3252>

Solution: Customer Info



The screenshot shows an IDE window titled "Java-samples - CustomerInfo.java". The main editor displays the following Java code:

```
1 import java.util.Scanner;
2
3 public class CustomerInfo {
4     public static void main(String[] args) {
5         Scanner scan = new Scanner(System.in);
6         String fname = scan.nextLine();
7         String lname = scan.nextLine();
8         String email = scan.nextLine();
9         System.out.println("Customer: " + fname + " " + lname + " (" + email + ")");
10    }
11 }
```

The "Run" window at the bottom shows the command used to execute the program and its output:

```
2020.2.1\lib\idea_rt.jar=53695:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2020.2.1\bin" -Dfile
.encoding=UTF-8 -classpath C:\Projects\Java-samples\out\production\Java-samples CustomerInfo
Steve
Smith
steve@outlook.com
Customer: Steve Smith (steve@outlook.com)
```

- Write a program to read and print **info about a person**:
 - Reads 4 lines of **strings**: first name, last name, country and town
 - Prints information about a person in the following format:

"{firstName} {lastName} from {country} - {town}!"



Kelly
Smith
Ireland
Cork

The input data is presented in a light blue rectangular box with a thin black border. The text is left-aligned and consists of four lines: 'Kelly', 'Smith', 'Ireland', and 'Cork'.



Kelly Smith from
Ireland - Cork!

The output data is presented in a light blue rectangular box with a thin black border. The text is left-aligned and consists of two lines: 'Kelly Smith from' and 'Ireland - Cork!'.

```
String firstName = scanner.nextLine();
String lastName = scanner.nextLine();
String country = scanner.nextLine();
String town = scanner.nextLine();
System.out.printf("%s %s from %s - %s!",
    firstName, lastName, country, town);
```

This solution uses **string formatting** with **placeholders**

- Submit your solution in the SoftUni judge:

<https://judge.softuni.org/Contests/Practice/Index/3252>

Problem: Town Info

- Write a program, which reads and print **info about a town**:
 - Reads **3 lines** of input: **name** (string), **population** and **area** (integers)
 - Prints information about a town in the following format:
"Town {name} has population of {population} and area {area} square km."

```
Berlin  
3675000  
984
```



```
Town Berlin has  
population of 3675000  
and area 984 square km.
```

```
String townName = scanner.nextLine();
int population = Integer.parseInt(scanner.nextLine());
int area = Integer.parseInt(scanner.nextLine());
System.out.printf(
    "Town %s has population of %d and area %d square km.",
    townName, population, area);
```

%s means "string"

%d means "integer"

- Submit your solution in the SoftUni judge:

<https://judge.softuni.org/Contests/Practice/Index/3252>

Learn By Doing!



- Join the **SoftUni "Learn To Code" Community**

<https://softuni.org>



- Access the **Free Coding Lessons**
- Get **Help** from the **Mentors**
- Meet the **Other Learners**

