# Lab: Intro to Java

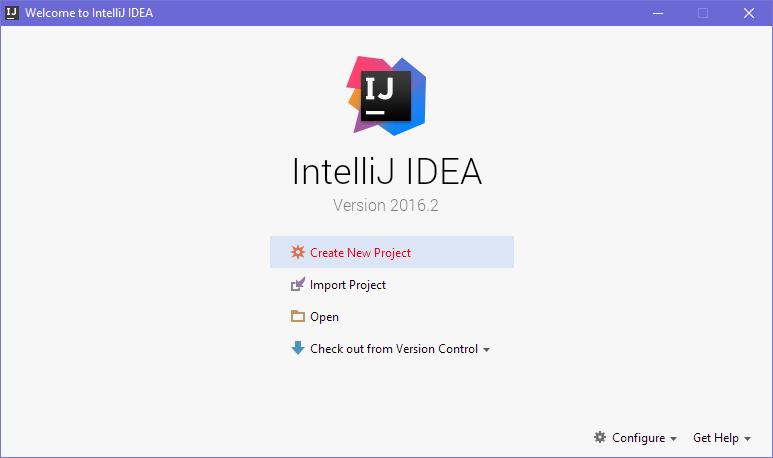
This document defines the exercises for ["Java Advanced" course @ Software University](https://softuni.bg/courses/java-advanced). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/382/).

# I/O and Data Types

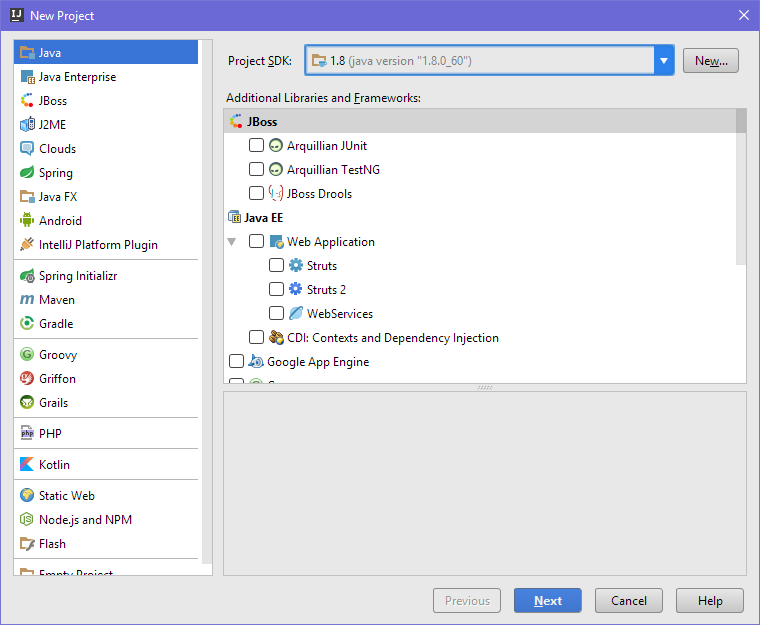
## Program "Hello Java"

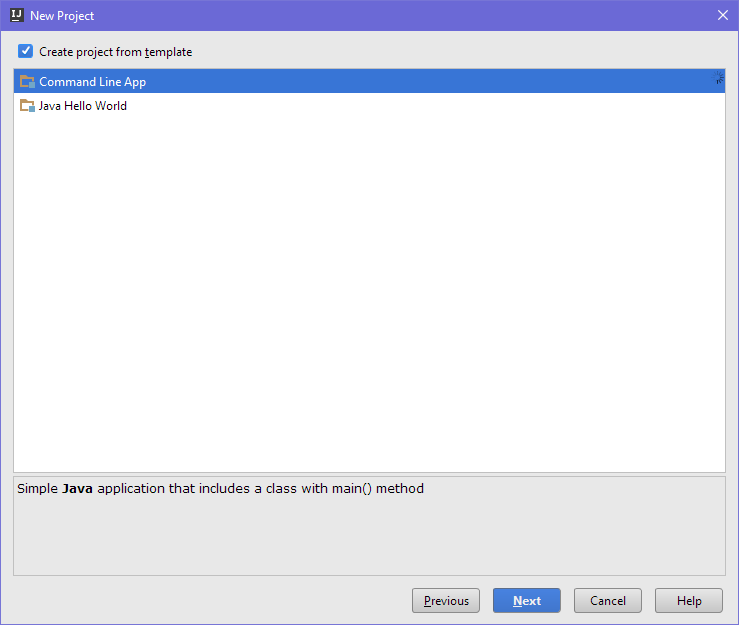
Write a console Java program, which prints "Hello Java".

1. Start IntelliJ IDEA.
2. Create new project: [Create New Project].

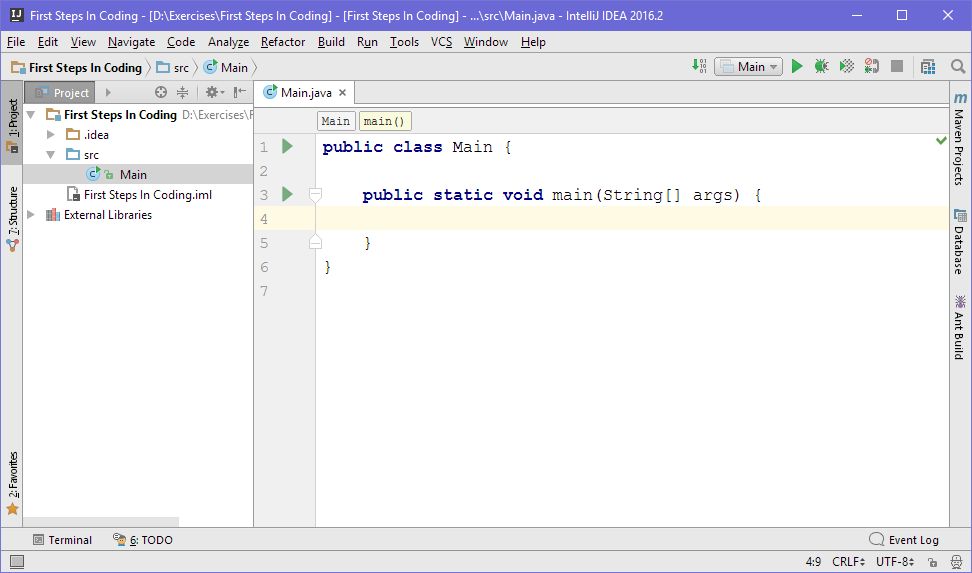


1. Choose [Java] 🡪 [Windows] 🡪 [Console Application] and give the project an appropriate name like "HelloJava":





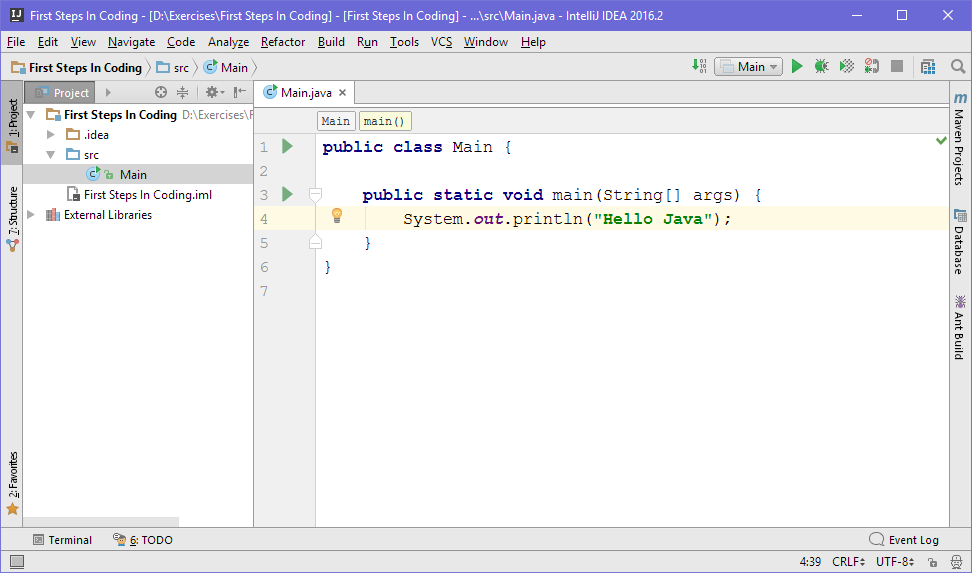
1. Find section main(String[] args). Write your Java statements there.
2. Place your cursor between the opening and closing brackets { }.
3. Press **[Enter]** after the opening bracket **{**.



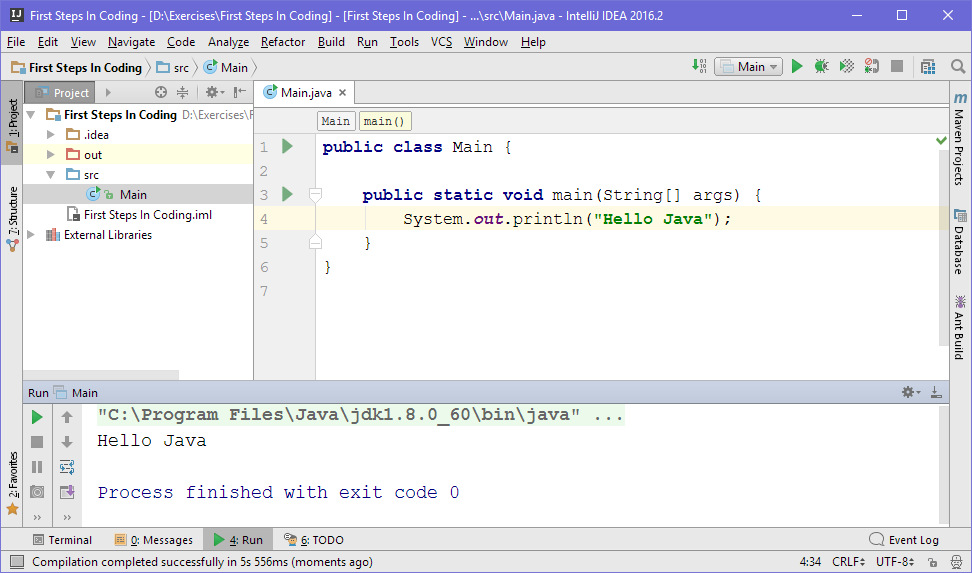
1. Write your statements (command for printing **"Hello Java"**):

|  |
| --- |
| System.out.println("Hello Java"); |

The code should be indented by a single tab.



1. Start the program by pressing **[Ctrl+Shift+F10]**. You should get the following result:



## Read Input

Write a program that reads:

* **Two strings** from the first line
* **Three Integers** which may be on multiple lines
* A **string** from the next line

Print the output in the following format **{firstWord} {secondWord} {thirdWord} {sumOfNumbers}.**

**The sum of numbers should be an integer.**

### Examples

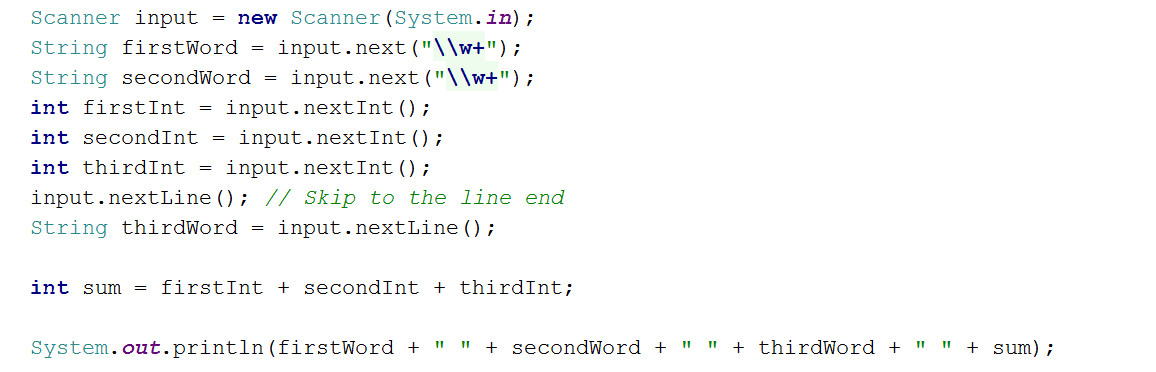
|  |  |
| --- | --- |
| **Input** | **Output** |
| Java Rocks  5 12 -7  End | Java Rocks End 10 |
| scanner system  1 2 3  in | scanner system in 6 |

### Hints

* Use the Scanner class
* Use the methods **next()**, **nextInt()**, **nextLine()**

### Solution

You might help yourself with the code below:



## Average of Three Numbers

Write program that **reads three numbers**. Print the **average of the three**, formatted to the **second digit after the decimal point.**

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2 4.5 3 | 3.17 |
| 3.1 4 15 | 7.37 |

### Hints

* You might use Scanner class
* To read the next double use the method **nextDouble()**:



## Euro Trip

You need to calculate the price of a given quantity of "wurst" in Deutsche Marks. Read the quantity as a double value and print the price in marks, given the following:

* The price of 1 kg wurst is 1.20 BGN
* The exchange rate is 4210500000000 DM : 1 BGN

Print the price, **rounded to the second digit** after the decimal separator.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2.35 | 11873610000000.00 marks |
| 1 | 5052600000000.00 marks |
| 15 | 75789000000000.00 marks |

### Hints

* Use the BigDecimal class to handle calculations involving money
* Don’t forget to format the output by using the method of the System class **printf()**.

## Greeting

Read two strings as an input – the first and last name of a person. Print a greeting in the following format:

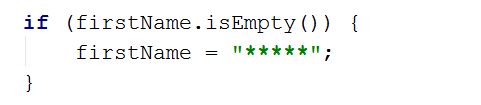
**"Hello, " + {firstName} {lastName} + "!"** , where if a name is missing, replace it with five stars "\*":

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Robert  Ford | Hello, Robert Ford! |
| Ford | Hello, \*\*\*\*\* Ford! |

### Hints

* Use the Scanner method **nextLine()**
* Use the String static method **isEmpty()** as in the example below:



# Conditional Statements and Loops

## Transport Price

A student travels **n** kilometers using only **one type of transport** based on the **distance that he will travel**:

* **Taxi**: Initial tax: **0.70 USD**. Daytime cost: **0.79 USD/km**. Night time cost: **0.90 USD/km**.
* **Bus**: Day / Night tariff: **0.09 USD/km**. For **at least 20 kilometers**.
* **Train**: Day / Night tariff: **0.06 USD/km**. For **at least 100 kilometers**.

Write a program that calculates the price of the trip by a given distance and time of day.

Format the output to the second digit after the decimal separator.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 5  day | 4.65 |
| 7  night | 7.00 |
| 25  day | 2.25 |
| 180  night | 10.80 |

## Numbers 0..9

Using a **while** **loop**, print the numbers from 0 to 9 inclusive.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
|  | Number: 0  Number: 1  Number: 2  Number: 3  Number: 4  Number: 5  Number: 6  Number: 7  Number: 8  Number: 9 |

## Product of Numbers [N..M]

Write a program that calculates the product of all numbers in the interval [**n**..**m**] by given **n** and **m**:

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 5 | product[1..5] = 120 |
| 3 20 | product[3..20] = 1216451004088320000 |

### Hints

* Use the **BigInteger** class to handle big numbers and a do-while loop.

## Lottery

Print all combinations from TOTO 3/10 lottery (like 6/49 but with less combinations):

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
|  | ...  ...  7 8 10  7 9 10  8 9 10 |

### Hints

* Use 3 nested loops.

# Bitwise Operations

## Extract Bit from Integer

Write a program that extracts from given positive integer **n** the value of given **bit at index** **p.** The bits are counted from **right to left**, starting from bit 0.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 5 2 | 1 | **n**=5, **p**=2  binary representation of **5**: 00000000 00000**1**01 |
| 0 9 | 0 | **n**=0, **p**=9  binary representation of **0**: 000000**0**0 00000000 |

## Modify a Bit

We are given a positive integer number **n**, a position **p** and a bit value **v** (v=0 or 1). Write a programthat modifies **n** to hold the value **v** at the position **p** from the binary representation of **n** while preserving all other bits in **n**.

Print the decimal representation of the resulting number.

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Comments** |
| 5 2 0 | 1 | **n**=5, **p**=2, **v**=0  binary representation of **5**: 00000000 00000**1**01  binary representation of the result (**1**): 00000000 00000**0**01 |
| 0 9 1 | 512 | **n**=0, **p**=9, **v**=1  binary representation of **0**: 000000**0**0 00000000  binary representation of the result (**512**): 000000**1**0 00000000 |