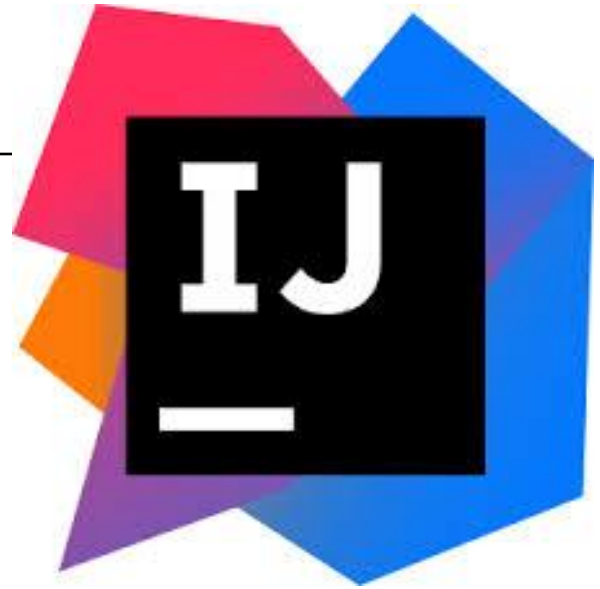
Programming Fundamentals
Week 1 Talk a

Introduction

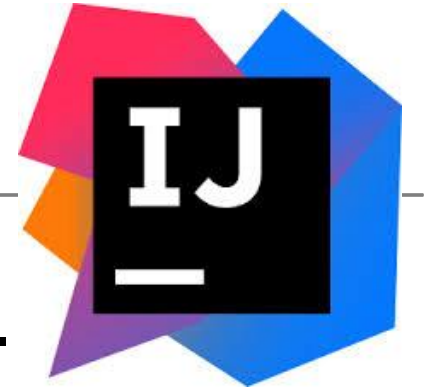
Produced by:
Siobhan Roche
Mairead Meagher
Peter Windle



Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University
SETU

Department of Computing and Mathematics
setu.ie

Learning to Program in Java



- Learn to think like a programmer (problem-solving, logic).
- Understand the building blocks: **classes, objects, methods, variables**.
- Use tools (IntelliJ, JVM) to write and run programs.
- Build from small programs → larger projects.

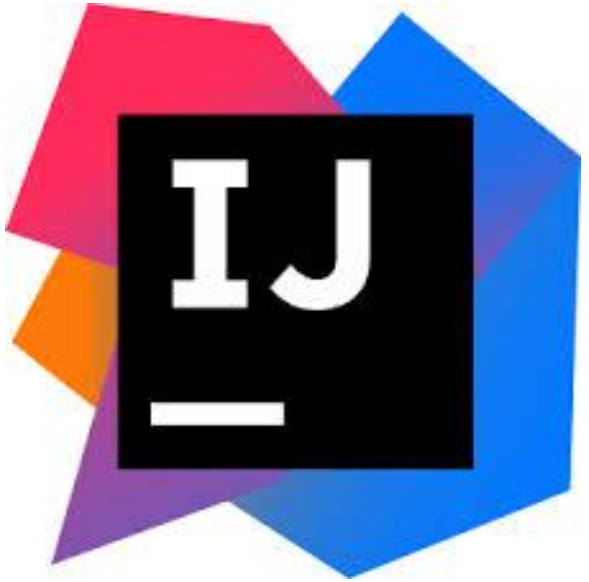
What You'll Be Able to Do…

- Write your own Java programs from scratch.
- Understand how code becomes a running program.
- Apply problem-solving skills to real-world examples.
- Gain skills valued by industry — Java is widely used in software development.

This Week..

- Getting started with Java
- Understanding JVM, main and IDE
- Write your first program

- Variables and data types
- Using Scanner class to enter information



Introduction to JVM

Java Virtual Machine and the main method

Produced by: Dr. Siobhán Drohan,
Ms. Mairead Meagher,
Ms. Siobhán Roche.

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

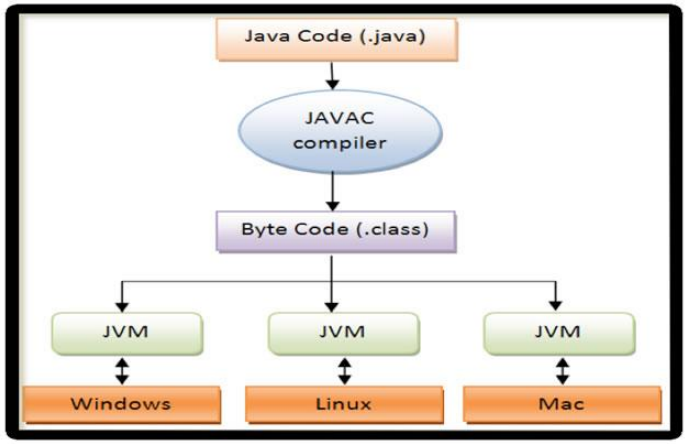
Department of Computing and Mathematics
http://www.setu.ie/

Topics list

1. Files in Java.
2. Java Virtual Machine (JVM).
3. main() method.
4. Why Java?

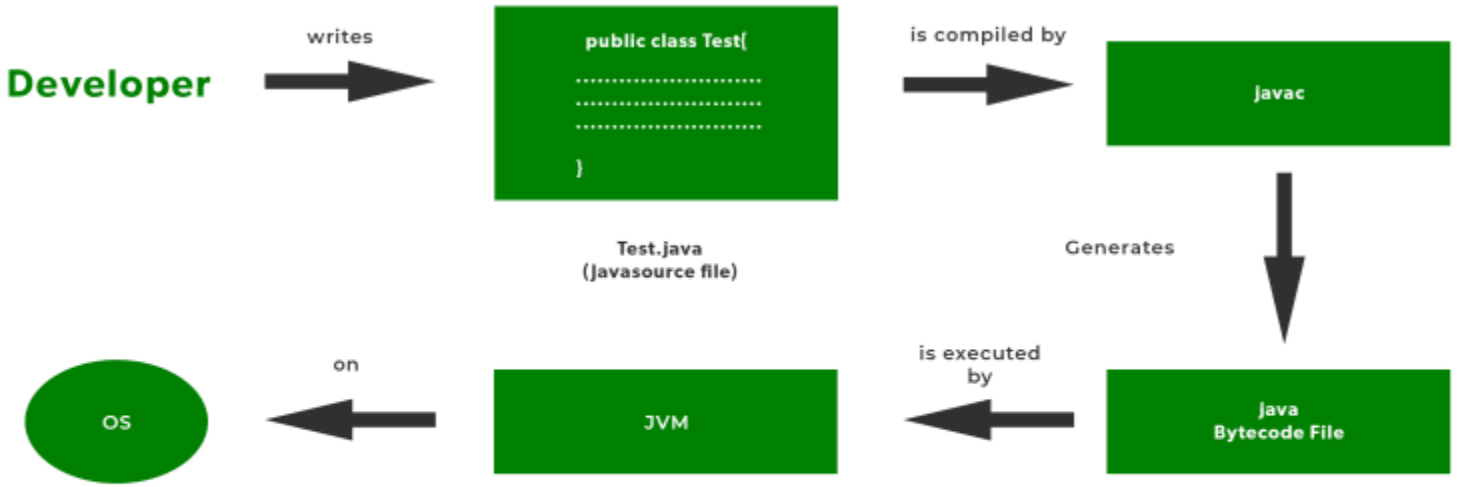
Files in Java

- **Java code** is written in **.java** file.
  - code contains one or more Java language constructs e.g. Classes, Methods, Variable, Objects etc.
- **Compiling** this code generates a **.class** file
    - A **.class** file
      - "**byte code**"
      - the input to Java Virtual Machine (JVM)
      - The **JVM** *(for a specific platform)*
        - **reads** this byte code
        - **interprets** it as machine code instructions *(for that platform)*
        - and **executes** the program *(on that platform)*



http://viralpatel.net/blogs/java-virtual-machine-an-inside-story/

Development Process



Step from java.code to machine code

- java source code is compiled.
- Transformed to bytecode by java compiler.
- Interpreted and executed by the java virtual machine on the underlying operating system.

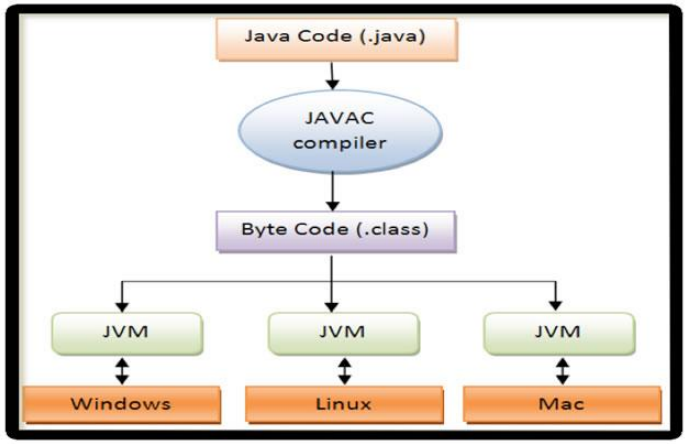
https://dev.to/leetcode/what-happens-when-we-run-a-java-code-1p8i

Topics list

1. Files in Java.
2. Java Virtual Machine (JVM).
3. main() method.
4. Why Java?

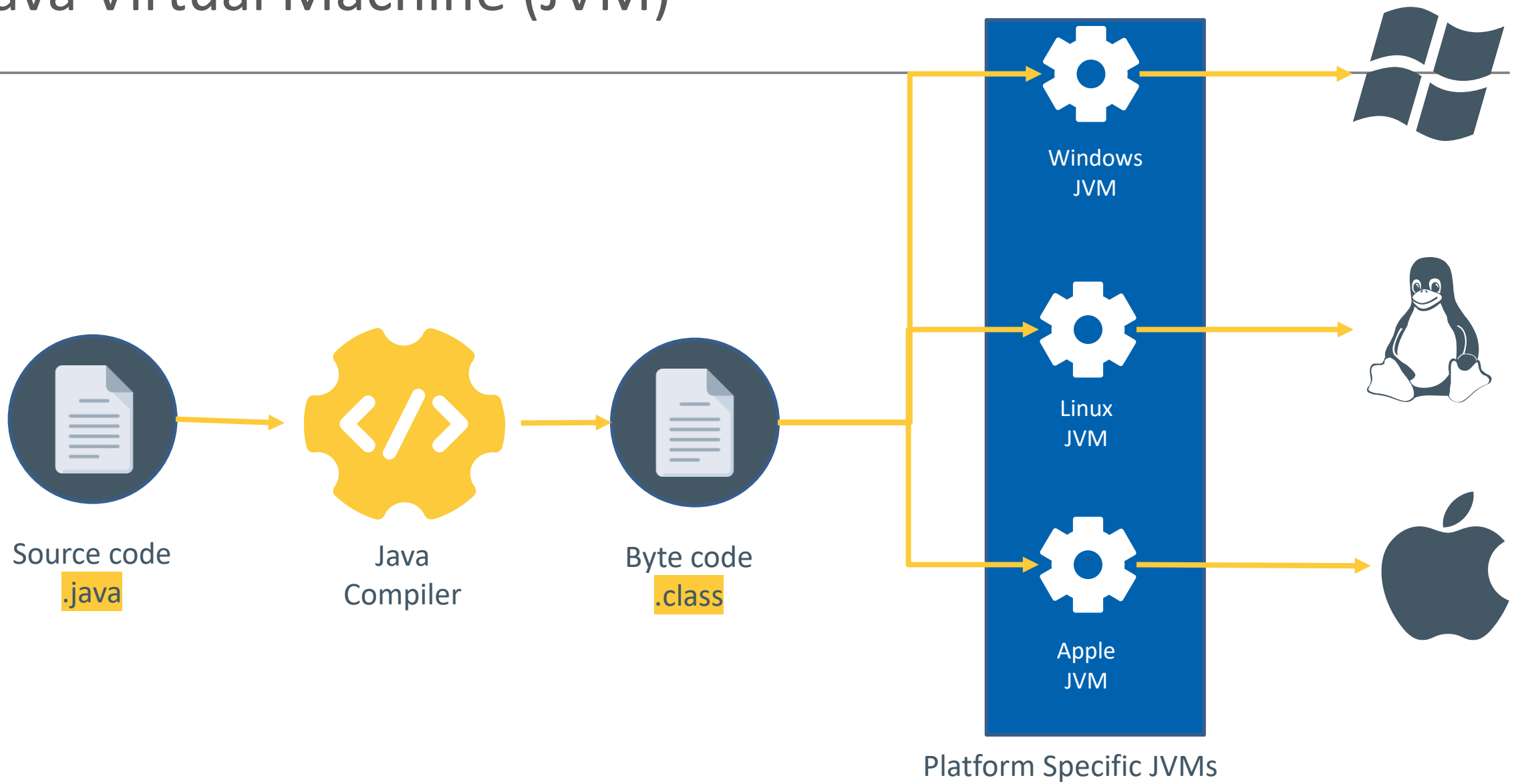
Java Virtual Machine (JVM)

- Java Virtual Machine (**JVM**)
  - is a "**virtual**" computer that resides in the "real" computer as a software process.
    - E.g. JVM for windows, for OSX, Linux etc.
- The JVM gives Java the flexibility of **platform independence**.
- The **.class** files can be run on any OS, once a JVM has been installed
  - (NB: JVM is installed when you install the JDK).



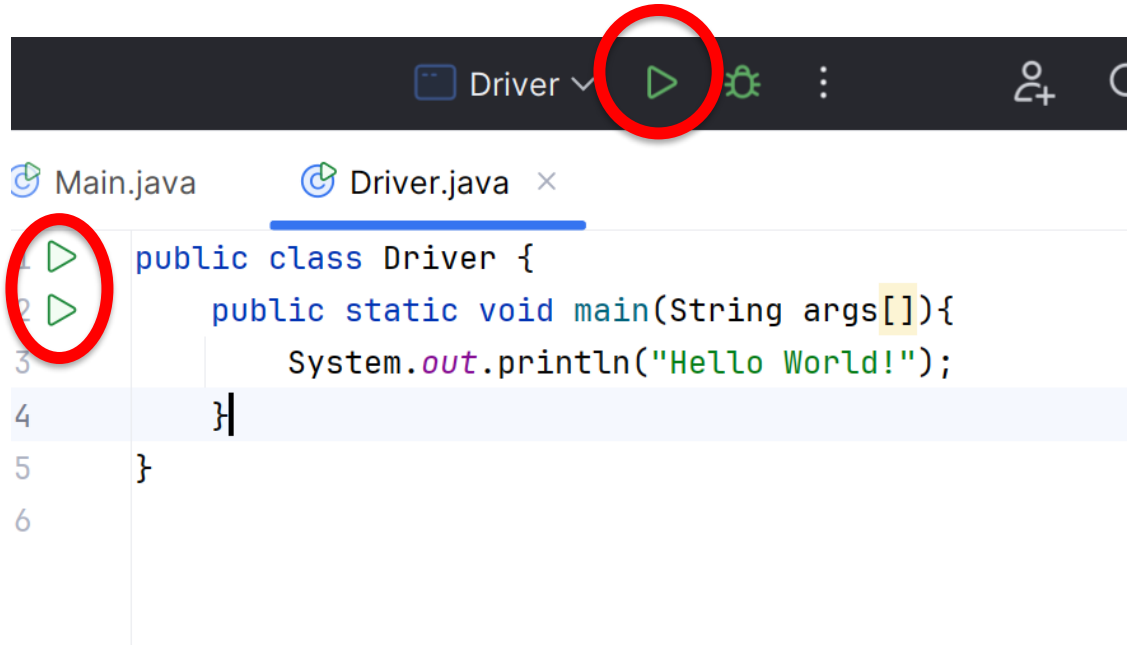
http://viralpatel.net/blogs/java-virtual-machine-an-inside-story/

Java Virtual Machine (JVM)



Example in IntelliJ

- When you click Run, IntelliJ compiles and sends bytecode to JVM
- JVM runs it on your system



Topics list

1. Files in Java.
2. Java Virtual Machine (JVM).
3. main() method.
4. Why Java?

main() method

- When you want to run a java project, the Java Virtual Machine (JVM) invokes the **main()** method in the project.

- For the JVM to recognise it, the main() method must have a specific method **signature**.

```
public static void main(String[] args)
{
   ...
}
```

main() method - **signature**

- “main” must exist
- “main” must be **public**
- “main” must be **static** (class method)
- “main” must have a **String array parameter**
- Only “main” can be invoked automatically.

```
public static void main(String[] args) {

    System.out.println("Hello world!");

}
```

Why This Signature?

- public: JVM must access it
- static: Belongs to class, not an object
- void: Doesn’t return a value
- String[] args: Can hold command-line input

Typical Program Flow

- Program starts in main()
- Create objects
- Call methods

```
public static void main(String[] args)
{
    Driver driver= new Driver();
    driver.runMenu();
}
```

Creates an object

Calls the first method

Putting It All Together

- Java code → .java → .class (bytecode)
- JVM = platform independence
- main() = starting point
- IntelliJ automates compiling + running

Topics list

1. Files in Java.
2. Java Virtual Machine (JVM).
3. main() method.
4. Why Java?

Why Java?

- Java is one of the most in-demand programming languages
- Java is a great language for beginners to break into engineering
  - Strong market for entry-level Java developers.
  - Strong computer science fundamentals
- The world's biggest and best companies use Java

Questions?

IntelliJ

Lets get started in our new IDE

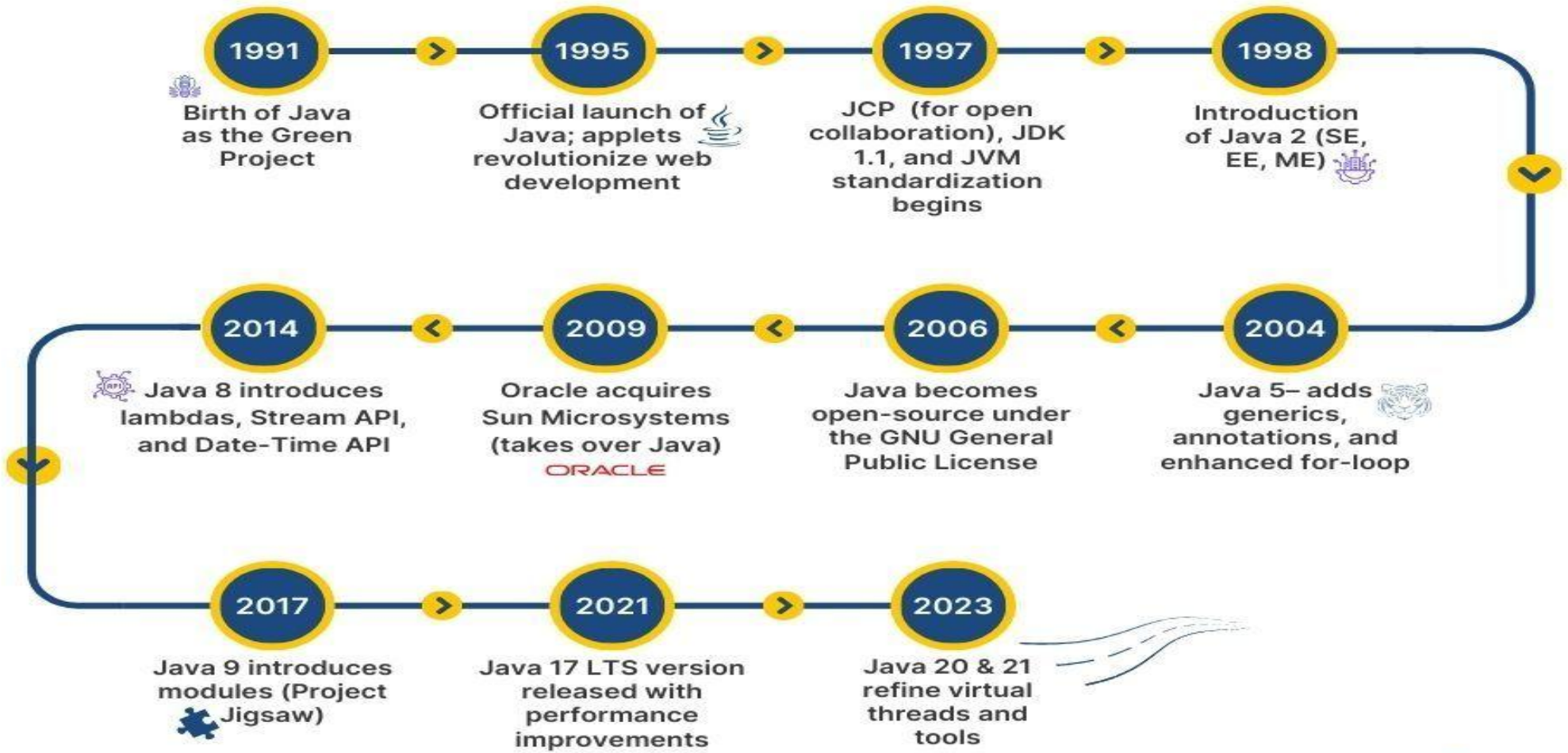
Produced by:

Dr. Siobhán Drohan,
Ms. Mairead Meagher,
Ms. Siobhán Roche.

Ollscoil Teicneolaíochta an Oirdheiscirt
South East Technological University

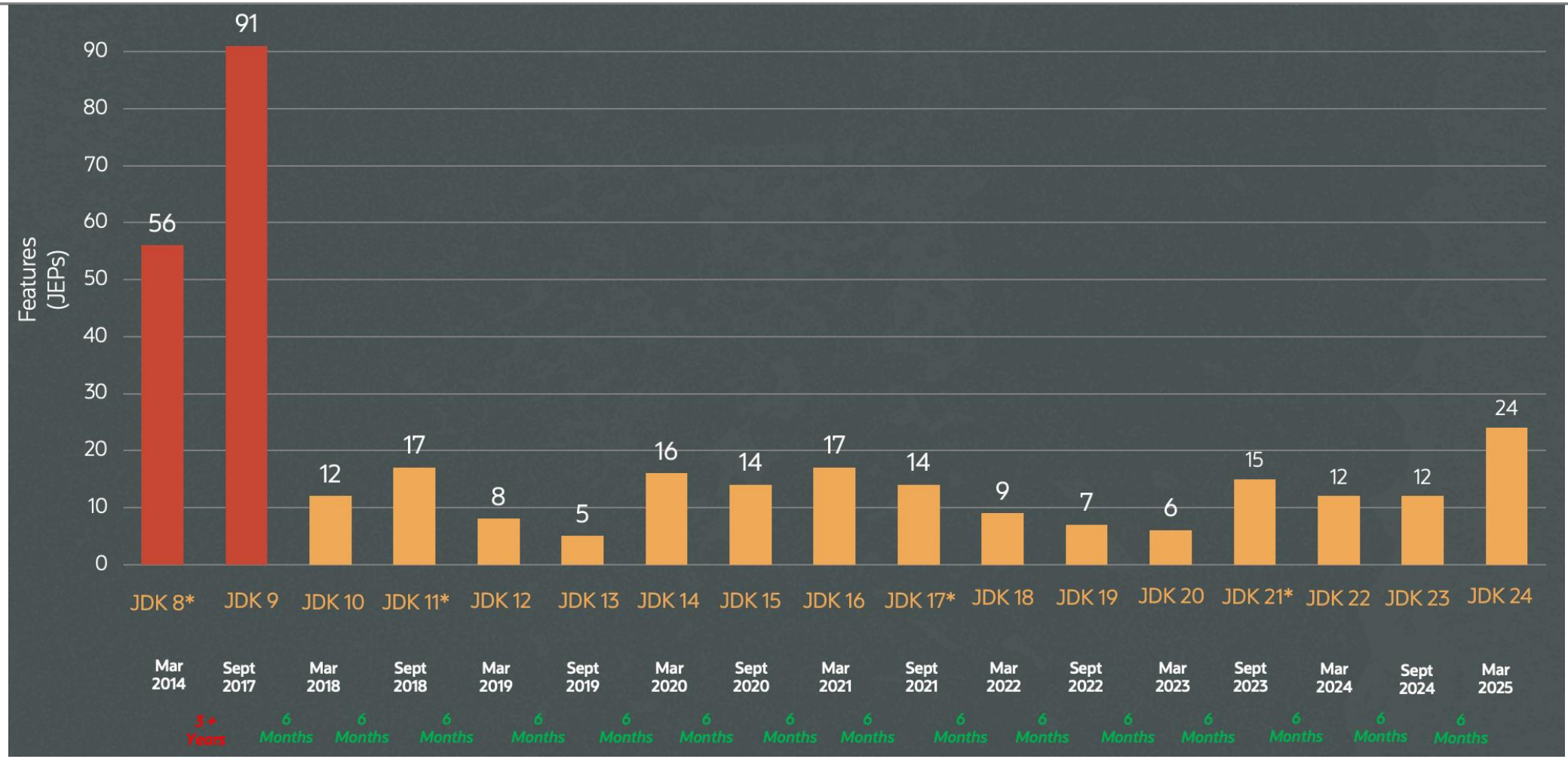
Department of Computing and Mathematics
http://www.setu.ie/

Timeline– History Of Java



unstop

JDK releases – 6 month cycle



What version of JDK should I install?

- The most up to date version that is available
  - Currently Java 25

What is an IDE?

- An IDE (Integrated Development Environment) is a software application that bundles all the necessary tools a programmer needs for software development into a single, easy-to-use graphical interface.
- An IDE (Integrated Development Environment) allows you to
  - quickly create applications by combining a source-code editor with the ability to compile and run your code,
  - integration with build, test and debug tools, version control systems, and so on.
  - search and navigate your codebase in ways your file system won't.

IntelliJ - An industry standard IDE

jetbrains.com/idea/download/?section=windows

JETBRAINS AI Developer Tools Team Tools Education Solutions Support Store

IntelliJ IDEA JetBrains IDEs Coming in 2025.3 What's New Features Resources Pricing Download

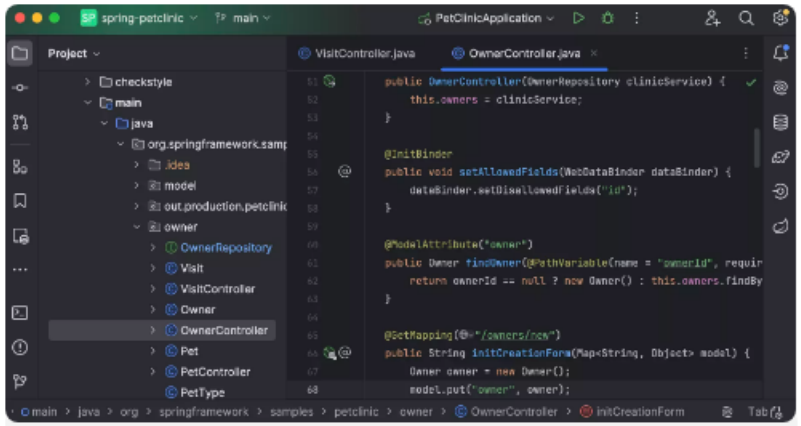
Windows macOS Linux

IntelliJ IDEA Ultimate

The Leading IDE for Professional Development in Java and Kotlin

Download .exe (Windows)

Free 30-day trial

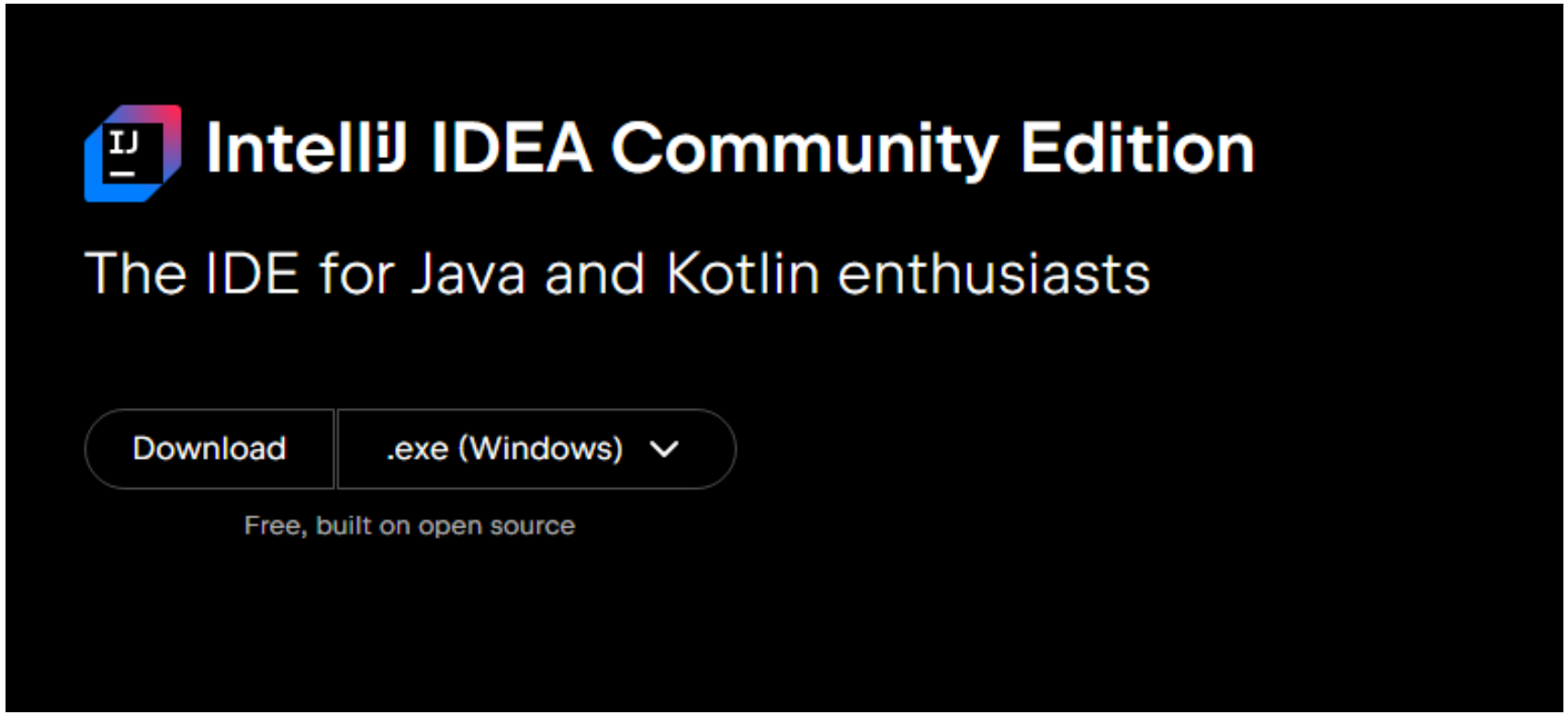


Version: 2025.2.2
Build: 252.26199.169
19 September 2025

System requirements Other versions
Installation instructions Third-party software

Community Edition

- You can scroll past the Ultimate version, and choose the free Community Version

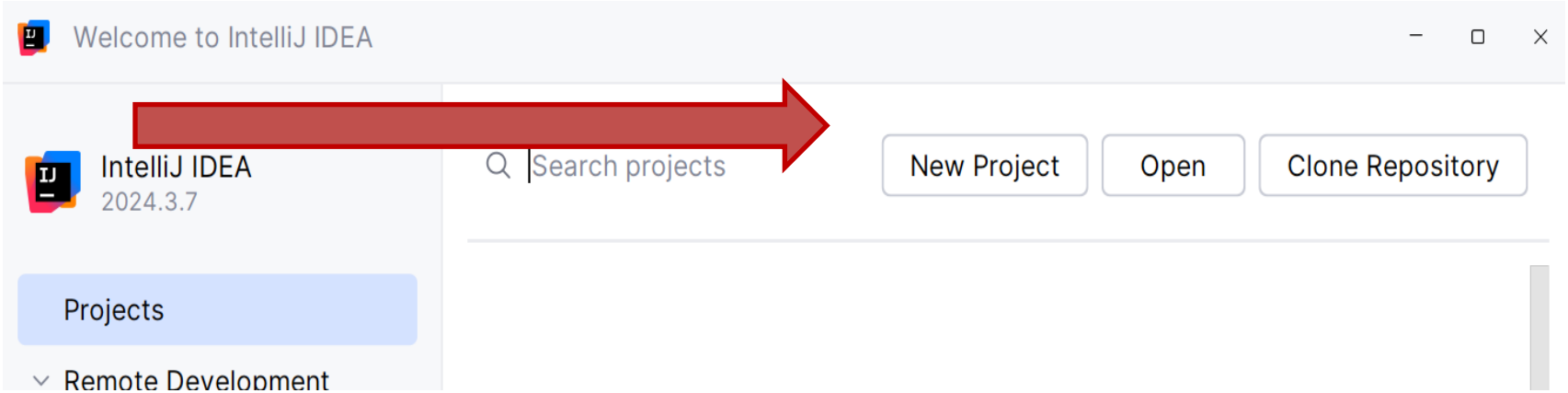


Hello World

IntelliJ

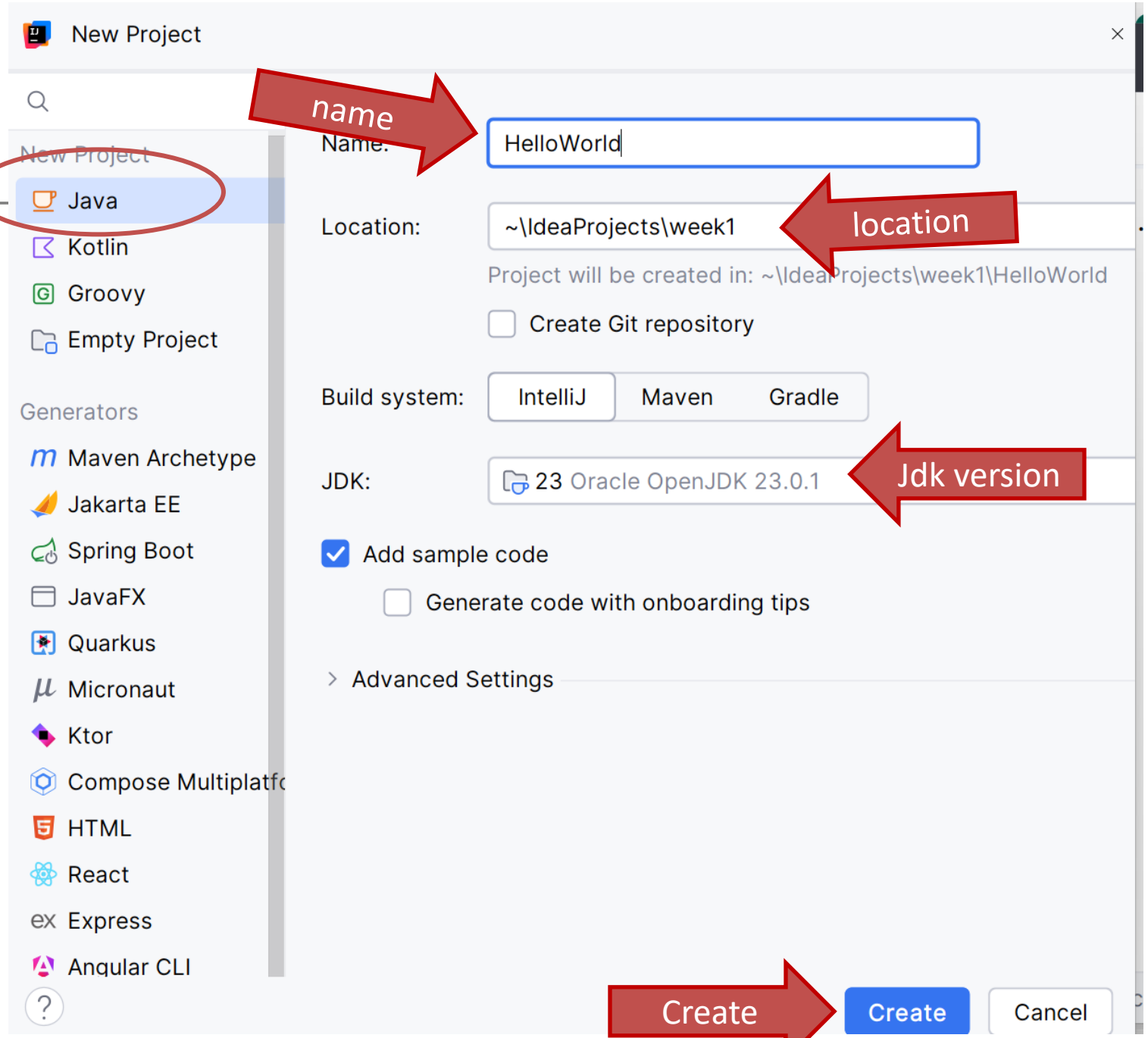
Launch IntelliJ

We can create a new project from the **Welcome** screen,

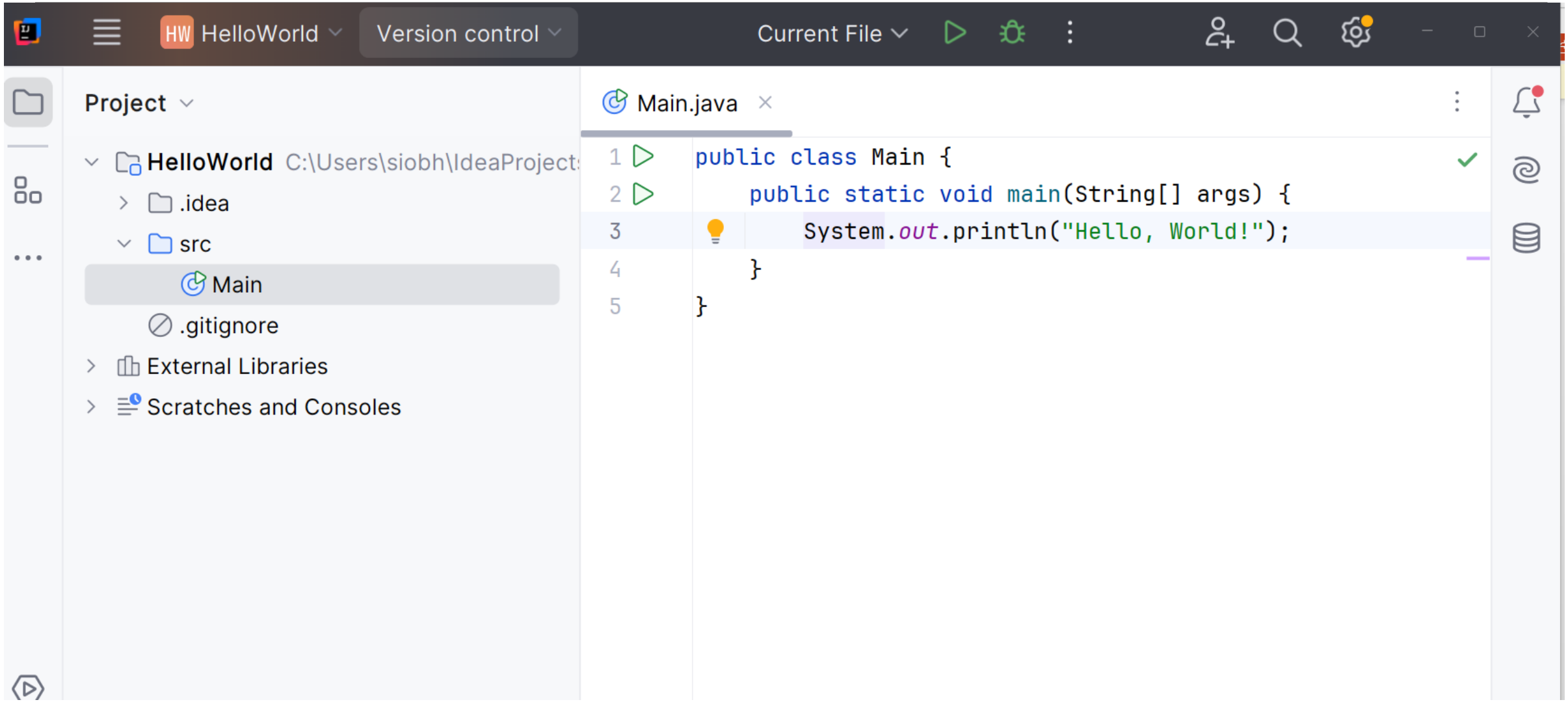


or we can go to **File | New | Project** in the main menu.

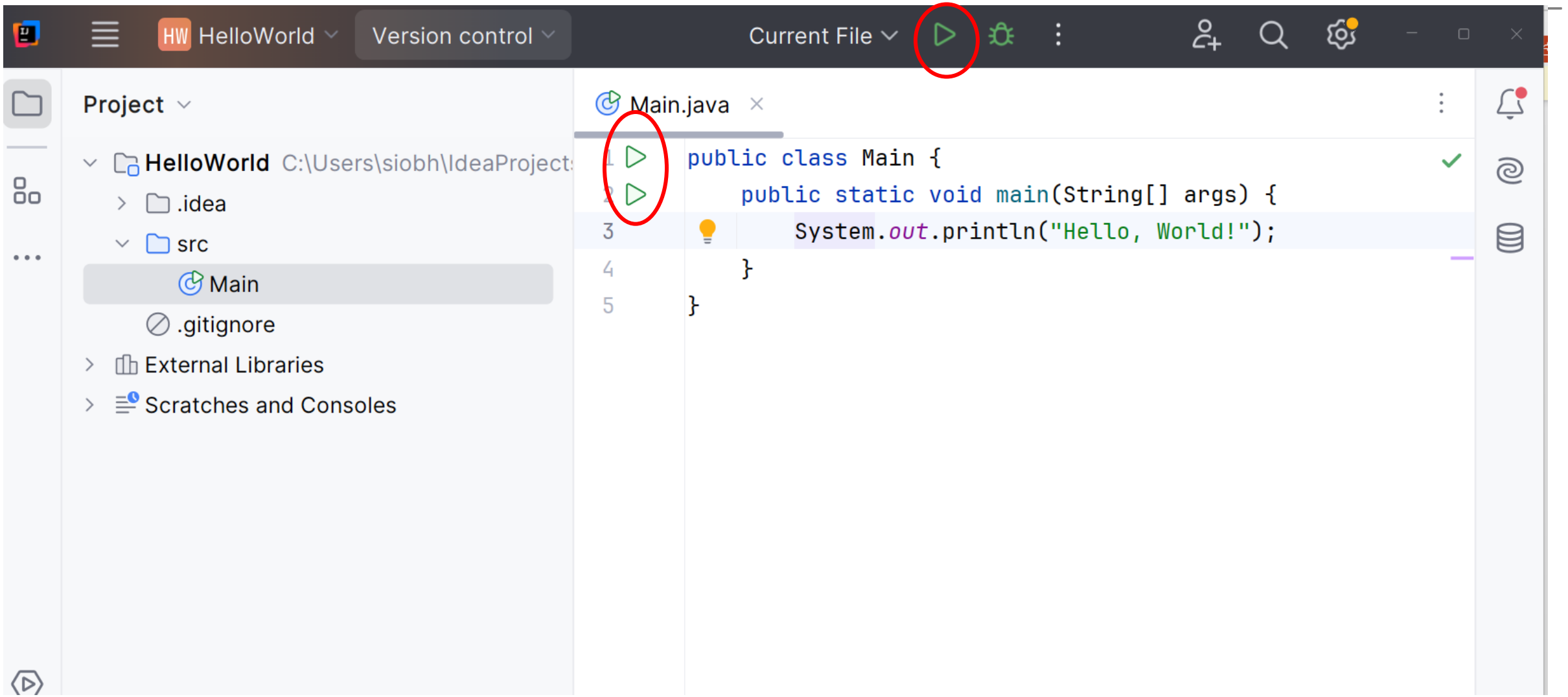
New Java Project



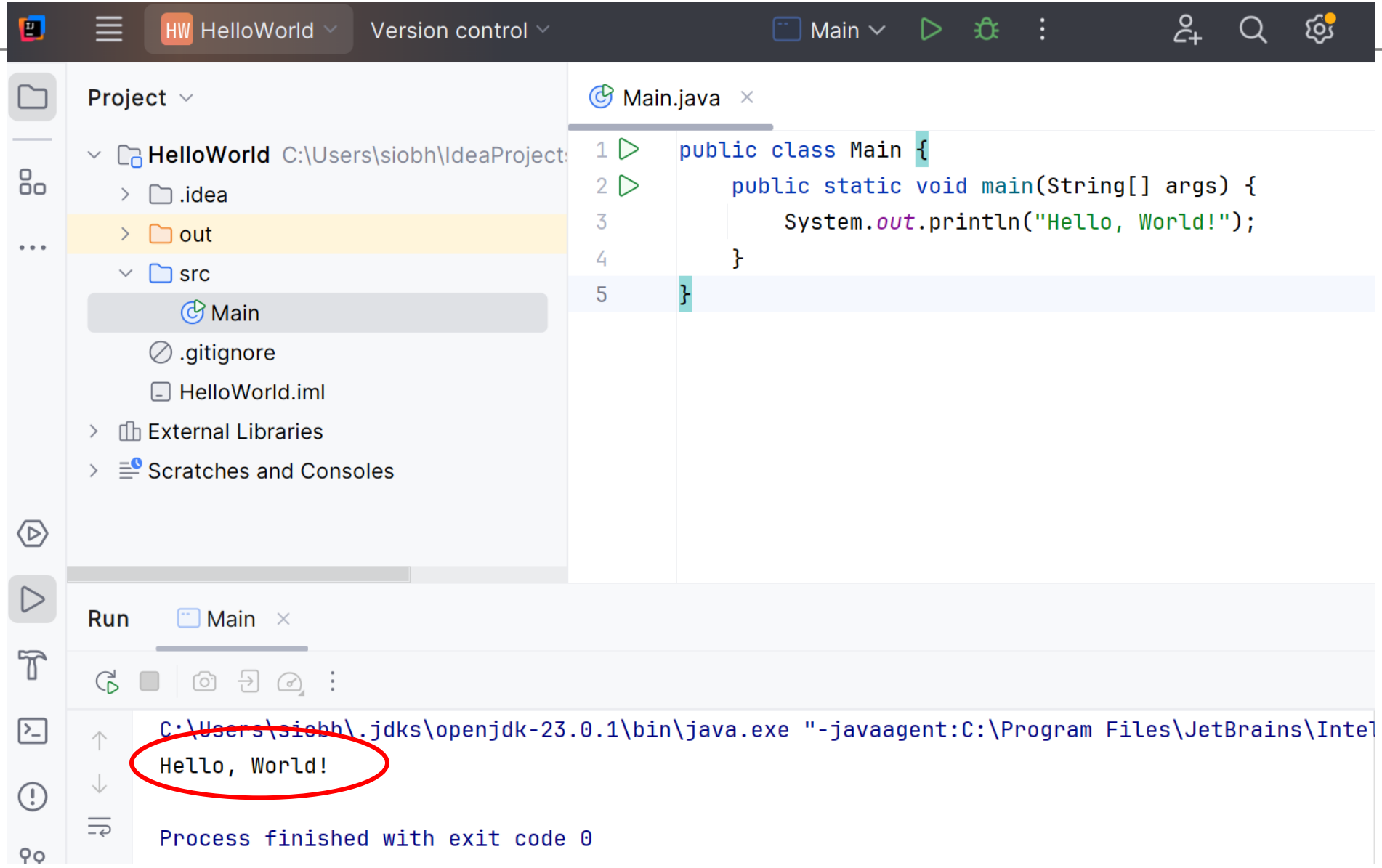
New Project – main method



Run app


HelloWorld
Version control
Current File
Project
HelloWorld C:\Users\siobh\IdeaProject
.idea
src
Main
.gitignore
External Libraries
Scratches and Consoles
Main.java
public class Main {
public static void main(String[] args) {
System.out.println("Hello, World!");
}
}

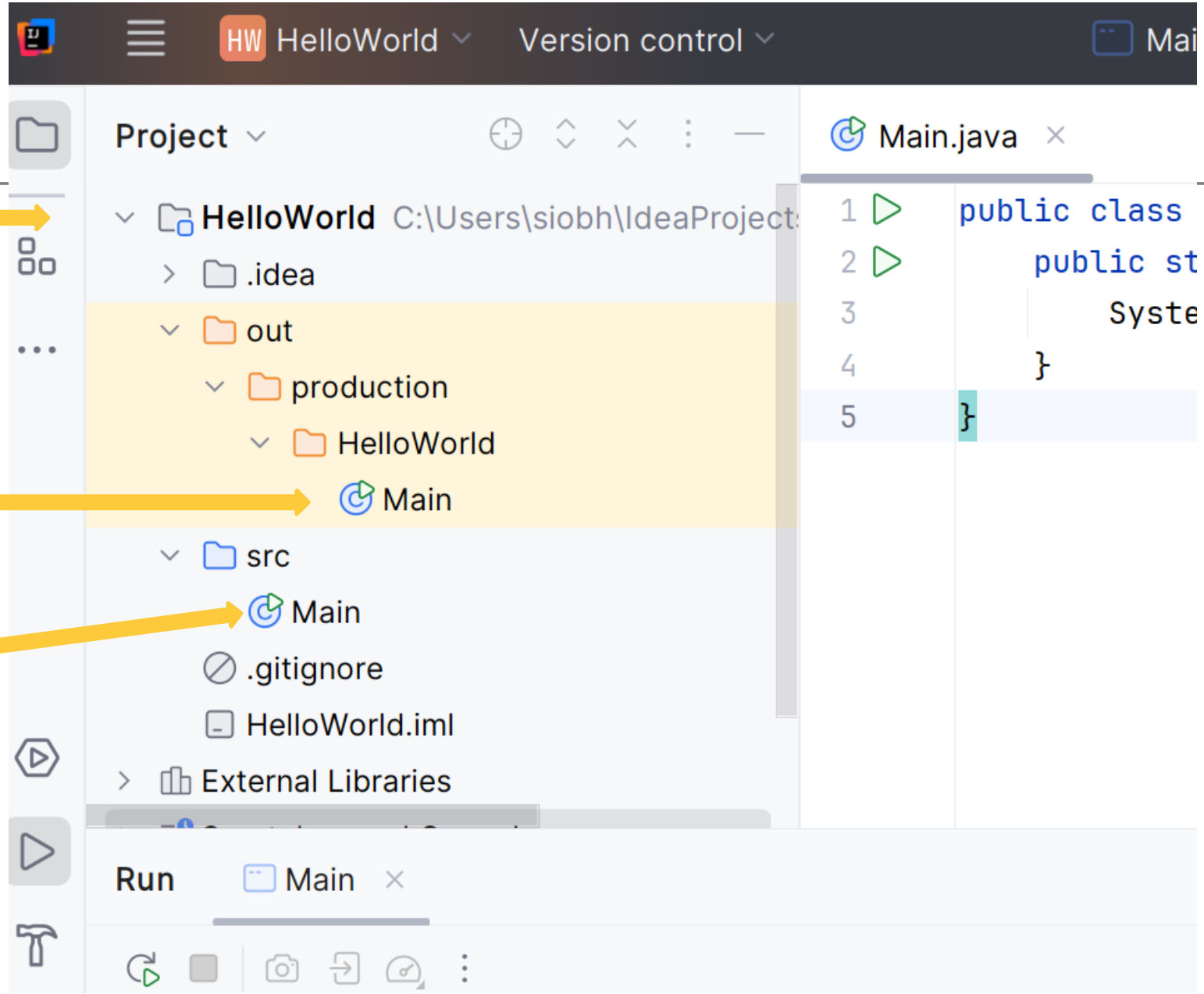
Console output



Project File

.class file
(compiled)

.java file
(source)



Lets get coding!

IntelliJ

Questions?