Get Going

ACTLEA
Active Learning

Learning objectives

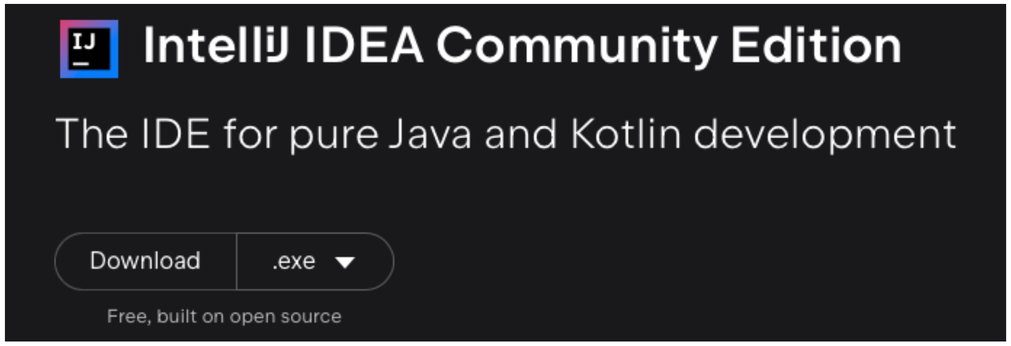
- Installation (of IntelliJ IDEA)
- Structure
- Classes and packages
- Naming conventions
- Comments

Working with the IDE (IntelliJ IDEA)

- An IDE (Integrated Development Environment) is used for programming
- IntelliJ IDEA is an IDE with many helpful tools for Java programming
- When running a program in IntelliJ, the source code will first be compiled to bytecode and then the bytecode will be run
- The debugger is really helpful when finding bugs!
- IntelliJ IDEA Community Edition is free and open source

Installing IntelliJ IDEA

- Download IntelliJ IDEA Community Edition for your operating system:
- https://www.jetbrains.com/idea/download/?section=windows
- Scroll down and download IntelliJ IDEA Community Edition:



- Double click the .exe file and follow the installation guide

ACTLEA
Active Learning

Demo 1 - Creating a program

- Creating a project in IntelliJ IDEA Community Edition
- Creating a Class
- Creating a main method
- Printing something to the console (output)

Exercise 1 - Hello World!

- Create a Hello World project in IntelliJ IDEA

- Hint:

```
public class Main {

    public static void main(String[] args) {
        System.out.println("Hello World!”);
    }
}
```

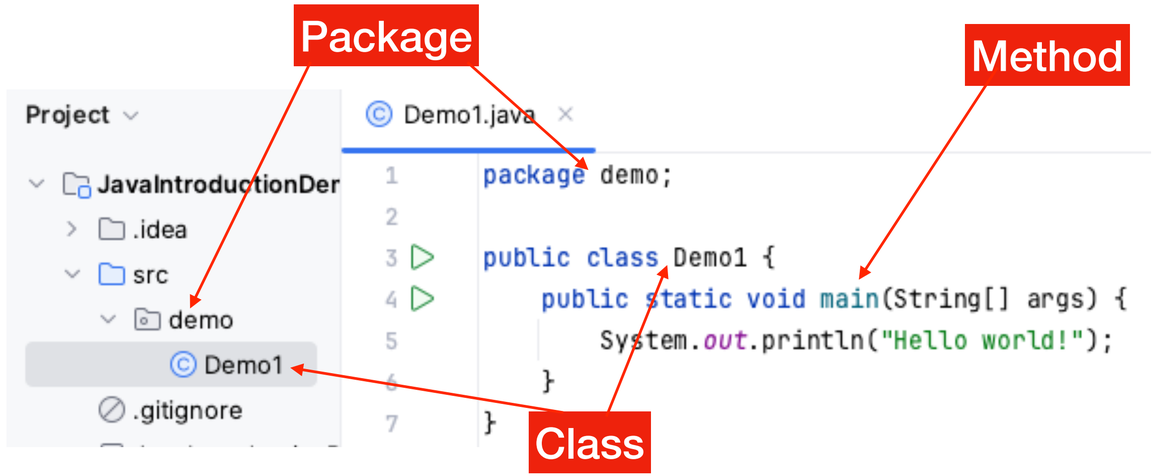
Exercise 2 - Stretch Tasks

- Experiment with the Hello World project:
- Print several lines by duplicating the line of code within the main method
- Try to use System.out.print instead of System.out.println, any difference?
- Can you duplicate the main method and have two methods in the class?
- What if you change the name in one of the methods?

Demo 2 - Structure

- IntelliJ IDEA structure

Basic structure of a Java program



- Classes - fundamental building blocks, contains the methods
- Packages - contains classes in a hierarchical structure
- Methods - functions containing Java code that can be executed

Packages

- Packages are used to organize and manage classes and prevent naming conflicts
- Related classes can be grouped within a package
- A package contains classes almost like a folder on a hard drive contain files
- Packages matter when it comes to Access modifiers

Classes

- A Java program typically consists of one or more classes
- Classes can contain methods and variables (behavior and data)
- Classes can also be used as blueprints for creating objects (OOP)

Demo 3 - Classes and Packages

- Creating and using packages
- Creating classes and organize classes in packages
- How packages can be used to avoid naming conflicts

Exercise 2 - Classes and packages

- Create two classes with the same name in the same project (both can contain a main method with the Hello World solution)
- There should not be any errors in the project and the main method of each class should be able to run successfully
- Hint: To avoid a naming conflict, put the classes in different packages

Java naming rules

- Names of packages, classes, methods and variables cannot contain spaces or certain special characters
- Names can contain numbers, but not start with a number
- Names cannot be reserved words (words with specific meaning in Java)

Java naming conventions

- Packages: Only lowercase letters (for example com.example.helloworld)
- Classes: CamelCase starting with an uppercase letter (for example Main, MyClass, HelloWorld, Demo1)
- Methods: camelCase starting with a lowercase letter (for example main, testMethod, getName, setAge)

CamelCase…

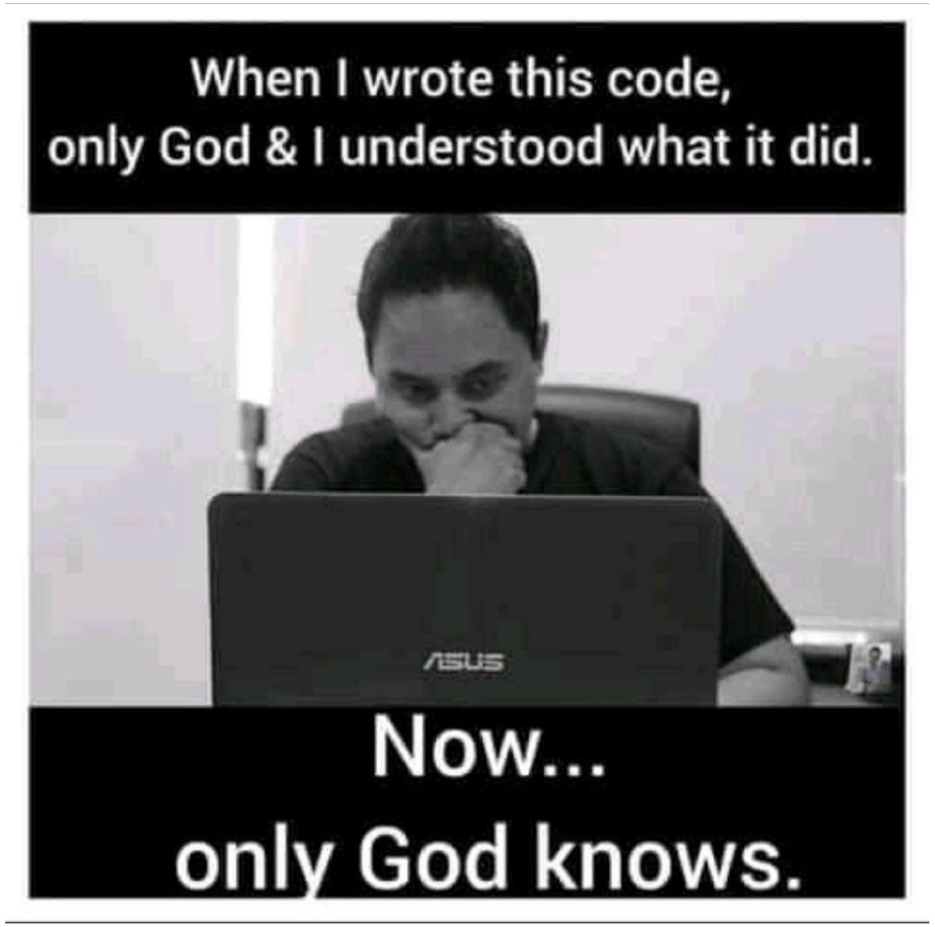
Comments

- Comments are notes within code that are ignored by the compiler
- They are used to explain the code, making it easier for developers
- Single-line comment with //
- Multi-line comment with /* and */
- Javadoc comment with /**, * and */

Java naming best practice

- The code should be easy to understand, both by others and by yourself in the future

- Descriptive can serve as a built-in decumentation, making the code self-explanatory and reducing the need for comments



Demo 4 - Comments

- Single-line comment with //
- Multi-line comment with /* and */
- Javadoc comment with /**, * and */

Learning objectives

- Installation (of IntelliJ IDEA)
- Structure
- Classes and packages
- Naming conventions
- Comments