

## Foundations

Stack → a data structure implementing pop from a list, pushing the popped element to another list.

Functional programming → describes the input processing method

Imperative programming → describes the exact command the machine executes

## JVM

In Java, we compile code that gets fed to the Java Virtual Machine, which has got a different machine code from the actual machine. It is also possible to simulate multiple processors on a JVM that runs on a single core.

This is because Java is meant to be machine-independent.

javac is the command to call the compiler, needed to compile the file.

After compiling, a .class file gets produced.

The object code is what is produced after compilation, that must be run by calling jvm + the filename without the extension (.class).

In IDEs, src (source) normally contains the code.

Each project can have different programs that can have different pieces of code. The actual program that can be run is the piece of code containing the "main" method. It's the method looked for by the compiler.

## Classes

Normally stored one for a file (that must have the same name as the class), their syntax is

```
public class <classname> {  
    body  
}
```

It is possible to write `public static void main` with `psvm`; it is possible to write, in a class:

1. Declaration of an `attribute` (passive object that store info)
2. Declaration of a `method` (active objects)
3. Declaration of the `main` method

All of these are `members` of the class. It is also possible to have `constructors`, `inner classes` (classes inside classes), `local classes` (classes inside methods).

The most important is the `main`, needed to run programs. It is possible to have `more than one main`. The order of the methods does not count.

Methods can have a `type` (public static `<type>`), receive `input` and `outputs` (`return ()`). Note that inputs need `type`. Since `main` has no `return type`, its `type` is `void`.

While  
Syntax:  
While (cond.) {  
}

In Java, it is possible to write ++ and -- near variable names, in order to change them by a single unit.

Useful commands:

- `System.out.println()`
- `""` for strings
- `/*` multi-line comment
- `*/`
- `//` single-line comment