Basic elements of a Java program

Structure of a Java program

Java is an object-oriented programming language, and this implies that we need to work with classes and objects. We'll see later what a class is, but, for now, we only need to know that every piece of code in Java needs to be placed inside a `class` clause.

1. Our first Java program

Let's see how to start with Java, by creating a simple Java program that prints "Hello" in the screen.

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

Let's explain this code:

- First line initializes the **class** in which we are going to place the code. We are creating a class called *MyClass*.
- Every code inside this *class* must be enclosed in **curly braces** `{` ... `}`
- Next, we start another piece of code called **main**, which is the main piece of code that will be launched when we run the program. In Java, this *main* block must always be *public*, *static* and *void*, and it must have a *String[]* set of parameters. We'll learn later what all this stuff means. Again, all the code belonging this *main* block must be enclosed in their corresponding curly braces.
- Finally, inside the curly braces of the *main* block, we add every instruction that we want to run. In this case, we are adding a **System.out.println** instruction, that is in charge of printing in the screen the text that we specify (*Hello*, in this case). Also, it's important to finish every instruction with a semicolon `;`. This tells the compiler that the instruction has finished, and we can start evaluating the next one. We could, this way, write more than one instruction per line, although this way of writing programs is not very usual.

2. More about this example

The structure of this program is very similar to the same program written in C#: we always need to define a class, even if we only need a `main` block. This `main` block is written in lower case in Java.

Besides, every public class must have the same name as the source file that contains it, so we need to store the source code of the example above in a file called `MyClass.java` (Java source files have `.java` extension). If we want to compile this code, we use `javac` tool from our JDK installation. We can do it through any IDE, such as Geany, or IntelliJ, as long as we have Java JDK properly installed.

```
javac MyClass.java
```

Then, a new file called `MyClass.class` will be generated. This is the compiled file that can be run under the *Java Virtual Machine* (JVM), using the `java` command. This last step can also be done under any IDE.

```
java MyClass
```

After running this program, we will see a "Hello" message in the screen.