Chapter 01: Building blocks

This chapter covers some basic concepts of object orientation (class members, what are instances, etc). It also covers some basic types and basic classes for data representation (dates, texts, etc).

Let's go then:

- Class members can be either attributes or methods;
- You define classes, methods and attributes by using Java keywords;
- Keywords inform the compile about scpecific characteristics you want something to have. For instance, a `public class Animal {}` code says to the compiler that `Animal` is a `class`, and have all it's characterists, and is also `public`, therefore being accessible by other classes;
- Attributes must be defined at least by a type and a name;
- A method's signature is given by it's name and parameters types. In the book, the given example is:

```
public int numberVisitors(int month) {
  return 10;
}
```

In which the method's signature would be `numberVisitors(int)`; * A *top-level type* is a data structure that can be defined independently, inside it's own file. Classes are top-level :sunglasses:; * Only one type can be declared as public in a same file; * A public types' name **must** match it's file's name;

The *main()* method

> ***NOTE:*** It's fun how these years of complex frameworks that handle complex things for you and spare you from a lot of boilerplating, you can almost forget about this little guy and how important it is.

- Finally the *"Hello World"* code, we're 74 pages deep in the PDF and it's finally here. (Yes I implemented the example, cause I'm such a great student :books:);
- If there's any syntax error, the compiler will let you know;
- Now here's a list of rules that must be followed when trying to declare and run a class:
  - Only one **public** type per file;
  - The public type's name must match the file's name;
  - If the class is an entrypoint to the program, then it must contain a valid *main()* method (using the signature `main(String[])`);
- Dissecating the *main()* method:
  - `public`: Access modifier, tells who can call this method (`public` means any class in the program);
  - `static`: This modifier binds the following member to the class, making so that that member can be accessed by using the class name, instead

of a class instance;
- `void`: The return type. Since the *main()* method doesn't actually returns anything, then it's return type is void. The book states that is a good practice to use `void` as return type for methods that intend to change some object's state, and since this is literally the purpose of the main method (change the program's state from started to finished), then is logic that you're going to use it as the return type here;

NOTE: I got a little curous about the declaration made by the authors above, because I have never really though about it that way. So I played a little with the main method by myself in the example's directory. All the classes I wrote are there, with comments saying either they worked or not. Feel free to try.

Stopped at page 10, playing around with the *main()* method's parameters.