**Object Oriented programming**

# Jey concepts of OOP

* Abstraction
* Encapsulation
* Inheritance
* Polymorphism

# Abstraction

This is arguably the most important [key concept](https://stackify.com/oops-concepts-in-java/) of object-oriented programming (OOP). It’s a concept in OOP that provides only the essential information. The rest is hidden from view.

# Encapsulation

By definition, encapsulation describes the idea of bundling data and methods that work on that data within one unit, like a class in Java. This concept is also often used to hide the internal representation, or state of an object from the outside. This is called information hiding.

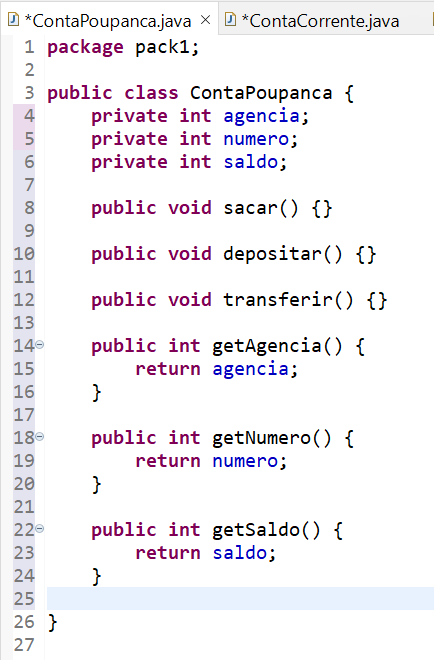
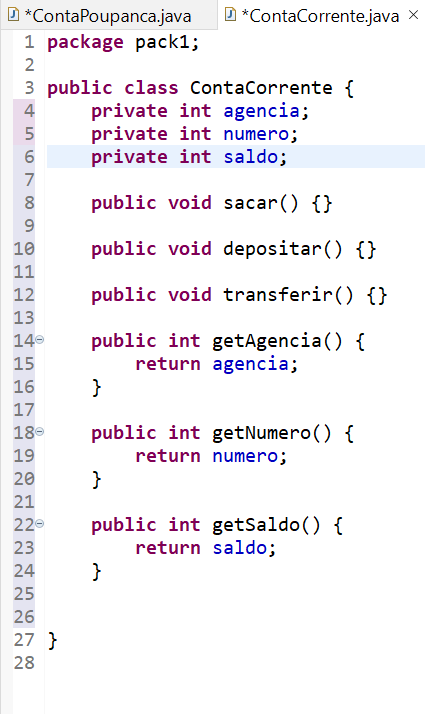
Another way to think about encapsulation is, it is a protective shield that prevents the data from being accessed by the code outside this shield.

# Inheritance

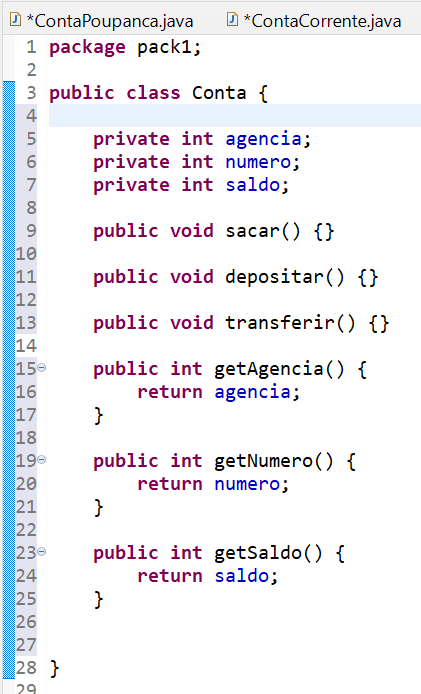
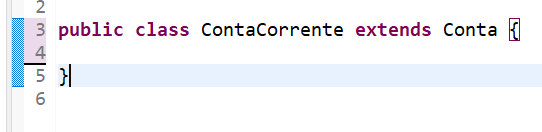
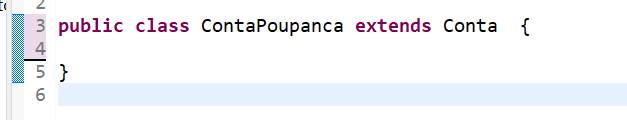
Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system). The idea behind inheritance in Java is that you can create new classes that are built upon existing classes.

In the examples, we can see a lot of code duplication. Both classes share the same attributes. This leads to more rework, as any changes have to be made to both classes.

DRY – don’t repeat yourself



A simple and elegant solution to that is to use inheritance.



# Polymorphism

Polymorphism is one of the core concepts of object-oriented programming (OOP) and describes situations in which something occurs in several different forms. In computer science, it describes the concept that you can access objects of different types through the same interface. Each type can provide its own independent implementation of this interface.

To know whether an object is polymorphic, you can perform a simple test. If the object successfully passes multiple is-a or instanceof tests, it’s polymorphic. As described in our post about inheritance, all Java classes extend the class Object. Due to this, all objects in Java are polymorphic because they pass at least two instanceof checks.