This is a project of a shop for the work of Object Oriented Programming.

The goal of this project is to create a small system of a shop, with a manager, an Assistant, a customer, products and prices. In this ecosystem, we can register people, consult the balance in the bank, change the function and salary of the Assistant, pay the employees, discount a specific amount from the Assistants, consult the cash of the shop, see the client's money, withdraw money, see the products of the shop, consult the prices of the products and buy products.

Let's look separately at the features and methods of each of the six classes:
[**Main**], [**Person]**, [**Manager**], [**Assistant]**, [**Store]** and [**Customer]**:

The [**Main**] class is the main class of the project, where the instantiations take place.

The class [**Person**] is a superclass of the classes [**Manager]**, [**Assistant**] and [**Customer]**.
In this single-method class, we can register the person with the following variables:

- *name*
- *dateBirth*
- *phone*
- *email*
- *accountBank*

The class [**Manager**] is a subclass of [**Person**]. This class has 3 specific variables with defined values, *salary*, *function* and *balanceBank*. In addition, it has the methods seeBalance, manageFunction, manageWage, payWage and discount.
With these methods we can, view the current balance in the bank account, change the role of the Assistant, change the salary of the Assistant, pay the salary of all employees and discount some amount from the Assistants' account. Something to consider in this class is that it doesn't have the Setter method of the *saldoBanco* variable, *because it can't change its balance.*

The [**Assistant**] class is a subclass of [**Person**]. This class is similar to the [Manager] class, having variables with the same names, however, there is a difference in the methods, this class doesn't have setters, since the idea here is that it can't change any of its variables (role, salary and balance) is the manager that must do it. The only methods in this class are the getters and the *verSaldo*, to view the account balance.

The [**Store**] class is the most different class in this project, it does not participate in inheritance. This class has Getters and Setters methods for all its variables, apart from those, its method is the *seeCash*, which shows the current cash of the shop. Besides that, all its variables have defined values, the first one refers to the shop's cashier value, and the others refer to the shop's products value.

The class [**Customer**] is a subclass of [**Person**], but this in turn, has different variables and methods from the other two subclasses of its parent class. With only one specific

variable, *money,* this refers to the amount of money that the customer has on hand, this will be used in the next methods.
This class has the methods *seeMoney*, to visualize the amount of money available, *withdrawMoney, to* withdraw more money, *seeProducts, to* visualize all the available products, *consultPrice,* to consult the price of the products and *buy, to buy* some of the products.

Finally, we have the interface [**Employee**], which implements the method in common, *verSaldo*, between the classes [**Manager]** and [**Assistant**] thus implementing polymorphism in the project. Besides having the variables in common: *salary*, *function* and *balanceBank*.

Inside the code it has several comments explaining each part of the code, as well as highlighting the evaluation criteria.