

Object oriented programming

Classes and objects in Java - Additional exercises



Exercise 1

Create a project named **Stores**, and implement the following classes:

- An abstract class called `Store`. It will have an abstract method called `welcome`, two private fields (`cash` and `drinkPrice`, both *double*) and a constructor that will receive the `drinkPrice` and initialize `cash` to 0.0. It will also implement a method called `payDrinks(int numOfDrinks)` that will add to the `cash` variable the payment (number of drinks * price).
- A class called `LiquorStore` that will extend `Store` class. It will implement the `welcome` method to show the message "Welcome to the liquor store", and it will have a new private field called `tax` (an integer). Its constructor will receive the `drinkPrice` and `tax` values, calling the parent's constructor when necessary. This class will also override `payDrinks(int)` calling first the parent's method (pay without tax), and then adding the tax value to the cash field.
- A `Main` class with the `main` method. This method will:
 - Create a `LiquorStore` object with `drinkPrice` = 8.95€ and `tax` = 20%. Pay for 10 drinks and print the cash to see if its value is 107.40€ (print 2 decimal numbers).
 - Instantiate a `Store` using an anonymous class. The drink price will be 8.95€ and the `welcome` method will say "Welcome to anonymous store! Our drink price is XX€" (where XX will be the value of `drinkPrice` attribute, with 2 decimals). Call the `welcome` method and also pay for 10 drinks. Show the resulting cash (should be 89.50€).
- Implement getters and setters when needed

Exercise 2

Create a project named **KillEnemies**, and implement the following classes:

- Interface `Character`: With a method called `isEnemy()` that returns a boolean.
- Class `Friend`: Implements `Character`. `isEnemy()` returns false.
- Class `Enemy`: Implements `Character`. `isEnemy()` returns true. Also implements a method called `kill()` that shows this message: "Ahhhggg, you killed me, bastard!".
- Class `Main` containing the `main` method. You must create a list (`ArrayList`) of 10 `Character` objects (5 friends and 5 enemies), then, using `Collections.shuffle(List)`, randomize the order of the items. You have to travel through all characters checking if they are enemies, and if they are, you kill them (call `kill()` method).

An example of execution message would be:

```
Character 0 is a friend! :-)  
Character 1 is a friend! :-)  
Character 2 is an enemy! kill it!  
Ahhhggg, you killed me, bastard!  
...
```