

Java 15 - Checkpoint #2

Java Advanced

It is possible to implement constructors, setter, getters or any other additional method where it is considered necessary.

1. Declare a class Person with the following attributes:
 - name: String
 - hunger: int
 - thirst: int
2. Declare an interface Eatable with the following methods:
 - onEat(Person person): void
 - isPoisoned(): boolean
3. Declare an interface Drinkable with the following methods:
 - onDrink(Person person): void
4. Implement two methods eat(Eatable) and drink(Drinkable) on Person. Each of them modifies the hunger and thirst values, respectively, by calling the method of the interface. If an Eatable is poisoned, then an exception is thrown.
5. Declare the following classes representing food, implementing the correct interfaces:
 - Apple
 - Cola
 - Yogurt

Each edible or drinkable should modify the hunger and thirst attributes of the Person object.

Note: consider you can both drink yogurt or eat it with a spoon.

6. Implement a method haveLunch(...) in Person, which takes a Collection of Eatable objects and a Collection of Drinkable objects as parameters. The method calls eat() and drink() passing as parameters every Eatable and Drinkable object in the collections, respectively.
7. Implement a main method where:
 - A Person object is allocated.
 - An HashSet of 3 Eatables is allocated. One of them should be poisoned.
 - An HashMap of 2 Drinkable is allocated, using String objects as keys.
 - The haveLunch() method is called on the Person object, passing the two data structures previously allocated and converting them if necessary.

SQL

Given the following table:

```
Person(  
    taxCode VARCHAR(255) PRIMARY KEY,  
    firstName VARCHAR(64),  
    lastName VARCHAR(64),  
    age INTEGER,  
)
```

Write the following queries:

1. Insert a Person entry with random values.
2. Select all people younger than 18.
3. Select all people whose firstName is "Pippo".
4. Count how many people are older than 30.
5. Report the count of people that have a certain age, for each age value present in the table.