Programming III

Laboratory 4

Objectives

- Classes, interfaces, inheritance
- Collections
- Generics

Exercises

1. Create the following class hierarchy:

   a) A class *Airplane* that has the following attributes: producer, code, number of flights, fuel capacity

   b) From *Airplane* class derive classes *FightAriplain* that has the following characteristics: can or cannot camouflage and weapon capacity; *LineArplain* that has the following characteristic: maximum number of passengers and *EntertaimentAirplain* hat has the following characteristics: current owner and a list of previous owners.

   d) *Line* and *entertainment* airplanes implement also the interface *LuxuryOptions* that contains the information about the fact if a plain has: noise-cancelling headphones or personal touch screen TV for each client.

   Requirements:

   1. Create the class hierarchy described

   2. Create a list (`LinkedList` or `ArrayList`) of airplanes and display it.

   3. Create and display a map that contains the type of airplane and for each type count how many

   airplanes are in the list. Display the map in the following format: planeType number of plays displayed like and arrays of stars

   a) FightAirplane **

   b) LineArpline *****

   c) EntertainmentAirplane ***

   4. Display from the list of airplanes the ones that have noise canceling headphones

   5. Find and display the EntertaimentAirplain that had the most owners

   6. Display the average fuel capacity of all air-plains from the list, display the average fuel capacity of fight airplanes

2. Write a generic method to count the number of elements in a collection that have a specific property (for example, odd integers, prime numbers, palindromes).

3. Write a generic method to exchange the positions of two different elements in an array.

4. Design a class that acts as a library for the following kinds of media: book, video, and newspaper. Provide one version of the class that uses generics and one that does not. Feel free to use any additional APIs for storing and retrieving the media.

5. Joe Mocha is defining an interface Appendable that includes an append method. He then defines two classes, MyString and MyList, which both implement Appendable. He wants Java's type system to allow a MyString to be appended to a MyString, and a MyList to be appended to a MyList, but not MyString to a MyList, or a MyList to a MyString. Here is his definition of Appendable:

```
interface Appendable {
    Appendable append(Appendable a);
}
```

What is wrong with this definition? What is a correct one? Also write a definition for a classes MyString and MyList and uses the revised definition of Appendable.

[https://courses.cs.washington.edu/courses/cse341/18sp/java/mini1.pdf]

https://docs.oracle.com/javase/tutorial/java/generics/QandE/generics-answers.html

Supplementary Exercise

1. Create a class Zoo which has a name and a list of Animals. From the class Animal derive classes Elephant, Tiger, Lion, Flamingo and Strung. The Zoo is able to manage the food stock, so add attributes to the classes in order to offer to the user the flowing information:
    a. Reported related to necessary food on day/month / animal type
    b. Create a generic method that for an animal type displays all animals that belong to the Zoo
    c. Create the class diagram for the proposed solution and identify the relations between the class
    d. Document the project code