# Programming III

# Laboratory 4

## Objectives

- inheritance
- enumerations

## Exercises

1. Create the following classes:
   - an abstract base class named Figure that contains the declaration of two methods for calculating the perimeter and the area of a figure and a field storing the information if it is filled or not
   - two concrete classes Circle and Rectangle derived from Figure

Resolve the following requirements:

a) Create an array of objects of type Figure and display it
b) Display the area for the figures from the array that are circles and perimeter for rectangles
c) Add a class Trapezium that inherits Figure class and add at the figure class objects of type trapezium.

2. Create an abstract class X-Men that are game personages, having like attribute the age and a method display power points. Create two concrete types of X-men: Apocalypse and Magneto. A player can add to the base character a set of powers like: teleporting, telepathy, athlete, wings that came with some extra power points.

a) Discuss the following design possibilities
   a. For each combination of X-Men and power create subclass
   b. A X-Men contains an array of powers, the powers are defined like an enumeration
   c. Create an abstract class Power that extends X_Man and contains a member of type X-Man from which derive classes teleporting, telepathy, athlete, wings.
      i. Draw the class diagram
      ii. How does the following client code fits the class diagram description:

```
X-Men apo = new Apocalypse ();
apo = new wings (apo);
apo = new teleporting (apo);
System.out.println(apo.getPowerPoints());
```

b) Implement variant c)

3. Create an abstract class Animal that has the following attributes: age, name and the type of environment they prefer (water, earth, air). For environment attribute use an enumeration. From class Animal derive the classes Mammal, Birds and Reptiles. Mammal have like attribute the number of average children that can be born at one birth; Birds have like attribute average speed of a fly; Reptiles have like attribute the number of eggs they deposit. From each class derive some concrete classes (I.e. from manifers: cat, dog). Create an interface Ability that contains the following methods:

i. boolean canFly ()
ii. boolean canSwim ()
iii. boolean canClimb ()

Each concrete class implements the methods from the interface (e.g. A dog can swim but cannot fly and climb).

Resolve the following requirements:

a) Create an array of animals and display it
b) Find the number of animals that can fly from the array
c) Display the animals from the array that can climb and swim

Remark: For each requirement create a separate method

4. Propose a class hierarchy for the following problem. Suppose that you have to write a program that will keep track of all the accounts for a bank. There are a number of different kinds of accounts that the bank supports.

- Regular Account - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is no interest. There is a penalty of 10.00 if the balance falls below a minimum of 500.00.
- Interest Account - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is interest of 7% paid monthly. There is no minimum balance required.
- Checking Account - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is annual interest of 7% paid monthly. There is a penalty of 10.00 if the balance falls below a minimum of 100.00. There is a charge of 0.10 for each transaction.
- CD Account - This account charges a fee of which is the smaller of 10 or 10% of the balance at the end of the month. There is interest of 15% paid yearly. There is no minimum balance required, but if there is a withdrawal before 12 months have gone by there will be a penalty of 20% of the current balance.

Each of these accounts has a personal identification number (PIN) with it to provide protection.

Identify the classes attributes and methods. Write a test program to test the implemented methods.

For more details see:
https://cs.calvin.edu/activities/books/java/intro/1e/HandsOnJava/labs/11/exercise/index.html