République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur Et de la recherche Scientifique

Université Batna 2

Programmation orientée objet (TP)

Interfaces -- TP02 (suite) --

Exercise 01:

Create a simple Java program that implements a **Calculator** interface to calculate the areas and perimeter of different shapes and display their information. The program should:

- Define an **AreaCalculator** interface with the methods **calculateArea**, **calculatePerimeter**, and **displayInfo**. The first and second methods should take no arguments and return a double value representing the area and perimeter of the shape, the displayInfo method should display information about the shape.
- Create separate classes for **Rectangle**, **Circle**, and **Triangle** that implement the **AreaCalculator** interface. Each class should have appropriate **constructors** and an implementation of **calculateArea**, **calculatePerimeter** , and **displayInfo**.

Exercise 02:

Using the solution for Employee Exercise 02 (TP2-Abstract), create the **EmployeeInformation** interface. Within this interface, move the abstract method **calculateSalary()** from the Employee abstract class to the **EmployeeInformation** interface. Additionally, add two different methods to the EmployeeInformation interface: the first one, **performWork()**, represents the work done by an employee, and the second one, **calculateExperienceBonus()**, represents a mechanism to calculate a bonus based on an employee's years of experience. Make all the necessary changes to achieve the goal of this exercise, including adding attributes if needed.

Exercise 03 (Challenge):

Create a simple interface that allows an object to be saved to some sort of storage medium. Implement the following:

1.Saveable (interface)

It has two methods:

- **write**(), takes no arguments, and returns
a List containing objects of type String.

- **read**(), takes a List of type String and doesn't
return anything.

2. Player (class)

It has four fields:

- Two **Strings** called **name** and **weapon**.

- Two **ints** called **hitPoints** and **strength**.

A constructor :

that accepts a String (name) and two ints (hitPoints and strength).It initializes name, hitPoints and strength with the newly passed in values. It initializes weapon with the default weapon "Sword".

And eleven methods:

- **Getters** and **setters** for all four fields.

- **write()**, same as interface. Return a List of the
fields.

- **read()**, same as interface. Store the values in the List. Make sure the List is not null and the size() is greater than 0 before storing the values.

- **toString()**, Print the details of the object.

3. Monster (class)

It has three fields:

- One **String** called **name**

- Two **ints** called **hitPoints** and **strength**.

A constructor:

that accepts a String (name) and two ints (hitPoints and strength). It initialises name, hitPoints and strength with the newly passed in values.

And six methods:

- Only getters for the three fields.

- write(), same as interface. Return a List of the fields.

- **read()**, same as interface. Store the values in the List. Make sure the List is not null and the size() is greater than 0 before storing the values.

- **toString()**, Print the details of the object.

Hint

- Player and Monster need to implement **Saveable**.
- To use arrays, you need to import array.