Lab Work #04

CPL102: Object-Oriented Programming

SS2024

These exercises will cover the concept of abstraction class, interface, nested classes and exception handling:

1. Create an abstract class named "Shape" with an abstract method "perimeter()". Implement the "Shape" class in two subclasses named "Rectangle" and "Square". Implement the "perimeter()" method in each subclass to calculate the perimeter of the corresponding shape.
2. Create an abstract class named "Employee" with an abstract method "calculateSalary()". Implement the "Employee" class in two subclasses named "Manager" and "Developer". Implement the "calculateSalary()" method in each subclass to calculate the salary of the corresponding employee based on their position and experience.
3. Create an abstract class named "BankAccount" with an abstract method "withdraw()". Implement the "BankAccount" class in two subclasses named "CheckingAccount" and "SavingsAccount". Implement the "withdraw()" method in each subclass to withdraw money from the corresponding account.
4. Create an interface named "BankAccount" with a method named "deposit()". Implement the "BankAccount" interface in two classes named "CheckingAccount" and "SavingsAccount". Implement the "deposit()" method in each class to deposit money into the corresponding account.
5. Create two interfaces named "Swimmable" and "Flyable" with a method "swim()" and "fly()", respectively. Implement both interfaces in a class named "Bird". In the "Bird" class, implement the "swim()" method to throw a "Not implemented" exception and the "fly()" method to fly the bird.
6. Create three interfaces named "Flyable", "Swimmable", and "Walkable" with a method "move()". Implement all three interfaces in a class named "Animal". In the "Animal" class, implement the "move()" method to move the animal according to its capabilities. Then, create three subclasses named "Bird", "Fish", and "Dog" that extend the "Animal" class. Override the "move()" method in each subclass to reflect the unique way that each animal moves (e.g. the "Bird" can fly, the "Fish" can swim, and the "Dog" can walk). Finally, create a main method that creates an instance of each subclass and calls their "move()" methods to demonstrate their unique movements.

Compiled by Pema Galey, Tutor

7. Write a Java program that reads a list of integers from the user and computes the average of the integers. If the user enters an invalid input, your program should throw an exception and prompt the user to enter a valid input. Your program should continue to prompt the user until a valid input is entered.

   Requirements are:

   - Your program should prompt the user to enter the number of integers they want to enter
   - Your program should then prompt the user to enter each integer one by one
   - If the user enters an invalid input (i.e. non-integer), your program should throw a "NumberFormatException" and prompt the user to enter a valid input
   - After all the valid integers have been entered, your program should compute and print the average of the integers