Object Oriented Programming

Assignment 1

Due date: Monday, 14th March, 2022

Problem 1.

(The MyPointclass) Design a class named MyPoint to represent a point with x- and y-coordinates. The class contains:

The data fields x and y that represent the coordinates with getter methods.

A no-arg constructor that creates a point (0, 0).

An overloaded constructor that constructs a point with specified coordinates.

An overloaded copy-onstructor that constructs a point from another MyPoint object.

A method named distance that returns the distance from this point to a specified point of the MyPoint type.

An overloaded method named distance that returns the distance from this point to another point with specified x- and y-coordinates.

Draw the UML diagram for the class and then implement the class. Write a test program that creates the two points (0, 0) and (10, 30.5) and displays the distance between them.

[optional] Create a static method in the test class named isPolygon, which receives an array of MyPoint and returns true, if the points in the array form a polygon, otherwise returns false. The points in an array form a polygon, if the first point and the last point are the same.

Instructions:

1. Use shadowing parameters names in constructors.
2. Always use getter/setter methods to read/write class fields (even inside the class methods).

Problem 2.

(ATM machine) Use the Account class created in Lab to simulate an ATM machine. Add another field named id to the Account class and update the account class constructors and add getter/setter methods accordingly.

In the test class:

- Create ten accounts in an array with id 0, 1, . . . , 9, and initial balance $100.
- The system prompts the user to enter an id.
- If the id is entered incorrectly, ask the user to enter a correct id.
- Once an id is accepted, the main menu is displayed as shown in the sample run.

- You can enter a choice 1 for viewing the current balance, 2 for withdrawing money, 3 for depositing money, and 4 for exiting the main menu. Once you exit, the system will prompt for an id again. Thus, once the system starts, it will not stop.

Sample Run:

```
Enter an id: 4 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 100.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 2 Enter
Enter an amount to withdraw: 3 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 97.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 3 Enter
Enter an amount to deposit: 10 Enter

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 1 Enter
The balance is 107.0

Main menu
1: check balance
2: withdraw
3: deposit
4: exit
Enter a choice: 4 Enter

Enter an id:
```