TCP1201 Object-Oriented Programming and Data Structures

Lab02 Objects and Classes

Exercise 1: Account Class

Define an `Account` class that contains the following data and operations. Your class should be encapsulated.

a) An `int` data field named `id` for the account.
b) A `double` data field named `balance` for the account (default 100).
c) A `LocalDate` data field named `dateCreated` that stores the date on which the account is created.
d) A no-arg constructor.
e) An overloaded constructor that creates an account with the specified id and initial balance.
f) Accessors for all data fields.
g) A method named `deposit` that deposits a specified amount to the account. If a negative amount is provided, reject the deposit and display an error message.
h) A method named `withdraw` that withdraws a specified amount from the account. If there is insufficient balance in the account or negative amount is provided, reject the withdrawal and display an error message.

Write a program that:

i) Creates an Array of 3 Account objects.
   - For the first account, use no-arg constructor.
   - For the second account, use overloaded constructor (id=102, balance=2000).
   - For the third account, ask user input for id and balance.

j) Print the details of all accounts.
k) Ask user to input for an existing account id, display error if it is not found. Otherwise, ask user to enter the amount to deposit and the amount to withdraw from the account, print the balance. Try entering negative amount and withdrawing more money than the balance.

Sample run:

```
Create the 3rd account
Enter account id: 103
Enter balance   : RM3000

Account #1
Account id = 0
Created    = 2022-11-16
Balance    = RM100.00

Account #2
Account id = 102
Created    = 2022-11-16
Balance    = RM2000.00

Account #3
Account id = 103
Created    = 2022-11-16
Balance    = RM3000.00

Depost and Withdraw
Enter account id: 103
Enter amount to deposit: RM33
```

```
   Balance                = RM3033.00

   Enter amount to withdraw: RM1000
   Balance                 = RM2033.00
```

Exercise 2: UML Class Diagram

Create a UML Class Diagram for the Account class in the previous exercise.

Exercise 3: Java Documentation

Create a Java Documentation (javadoc) for the Account class in the previous exercise.

A JavaDoc uses the following doc comment to describe the **public class**, **public constructors**, and **public methods** in a .java file.

```
/**
 * Describe the public class, public constructors and public methods.
 * @param - a constructor or method that has a parameter should use
 *          @param tag to describe the parameter.
 * @return - a method that returns a value or object should use
 *           @return tag to describe the value or object it returns.
 */
```

1. Study String.java and its Java Documentation. Focus on the description, @param, and @return. Do you see the connection?

2. Write appropriate doc comment for the Account class, its public constructors, and its public methods

3. To produce JavaDoc for Account.java file, enter the following command in Command Prompt:

   ```
   javadoc -d html Account.java
   ```

4. To view the javadoc, open html/Account.html.