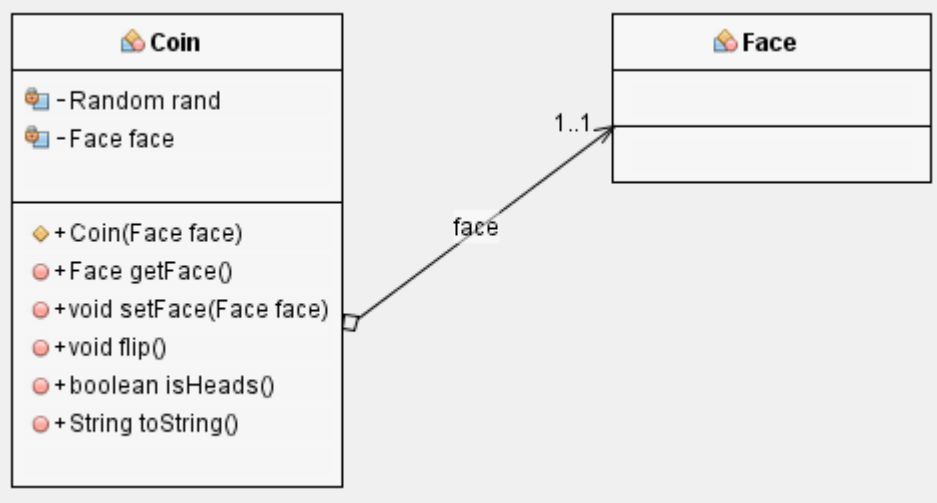
Sofia University
Department of Mathematics and Informatics

Course : AOOP part 1
Date: April 21, 2022
Student Name:

Homework No. 5

Problem No. 1

Given are class Coin and enum Face with constants HEAD and TAIL



Write a class called **MonetaryCoin** that is derived from the Coin class. Store an integer in the **MonetaryCoin** that represents its value and add a method that returns its value. Create a separate class **MonetaryCoinTest** with a main method to instantiate and compute the sum of several **MonetaryCoin** objects.Demonstrate that a monetary coin inherits its parent's ability to be flipped.

Submit the IntelliJ project. Use detailed comments and meaningful identifiers for variables and methods

Problem No. 2

Create a calculator as a **JavaFXML** **resuable visual** ***component*** in Scene Builder. The **Calculator** should allow the user to input numbers in a textbox and choose an operation to perform on them (addition, multiplication, division, subtraction) with **Buttons** as it is done with a usual calculator (*see the design of the* **Calculator** *application in the Accessories Program group in the MS Windows environment*) . Design the **Layout** of the buttons and the textbox to execute these operations, as well as, add support for handling the following events:

a) to remember the currently displayed number (**M** operation)

b) to add the currently displayed number with the number stored in memory and display the result (**M+** operation)

c) to subtract the currently displayed number with the number stored in memory and display the result (**M-** operation)

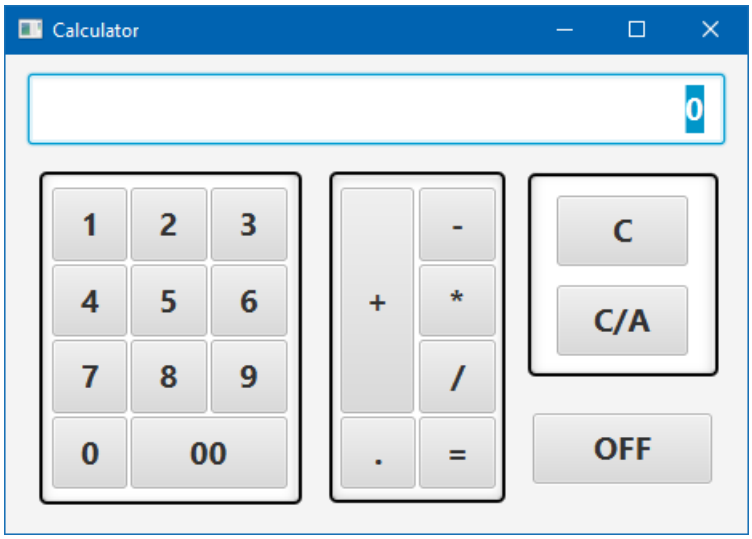
d) to clear the memory (**MC-** operation

Use a Stylesheet with Scene Builder to define the same GUI properties for the group of controls (buttons, textfield)

The methods performing the Calculator operations must be **public.** There should be also **two *public set* properties** for the user numeric input, necessary to complete the calculator operations**.** There should be a ***public get* property** for the Calculator result.

Package the reusable component as a JAR file and add it to the Scene Builder Library. The visual component must be accessible for *click- drag- and- drop* from the Custom section of Scene Builder

Write a modular JavaFXML application making use of the **reusable Calculator component**.



Submit the IntelliJ project. Use detailed comments and meaningful identifiers for variables and methods

Problem No. 3

Write a **Java Modular *FXML*** application that displays a **Registration form**. It collects the following data: (tool tip should display the required format in the respective text field)

a) the user **name**- it should be containing ONLY letters ***a- z*** (*upper* or *lower* case) and **at least 2** such letters

b) the user **phone** number- it should be of the format (**9999) (9999999**) i.e. a two groups of digits (4 and 7, respectively) surrounded by parenthesis.

c) The **email address** of the user- it should be **legal email address** (for instance, **check** for the following **at least**- *no duplicate @, no spaces inside, no duplicate dots (dots without anything in between*) etc.

d) **Password** and **Confirm password**- **both** entered **strings should be identical** upon validation

The above text fields should be **not left empty** (*indicate it on the form for the convenience of the user*) and **validated** using **regular expressions**

On errors in validating the user input, display respective Labels next to the wrong input.

Use a Stylesheet with Scene Builder to define the same GUI properties for the group of controls (buttons, textfield)

Submit the IntelliJ project. Use detailed comments and meaningful identifiers for variables and methods