Sofia University
Department of Mathematics and Informatics

Course : OO Programming with Java
Date: October 8, 2018
Student Name:

Lab No. 2a

Problem 1.

1. **Using the IntelliJ to create the UML model for**

   class Celsius:

   It has a double data member named *celsius*

   Add a constructor, set and get methods, as well as, a method

   double convertToFahrenheit() **using the following formula**

   ```
   fahrenheit  = 9.0 / 5.0 * celsius  + 32;
   ```

   class Fahrenheit :

   It has a double data member named *fahrenheit*

   Add a constructor, set and get methods, as well as, a method

   double convertToCelsius() **using the following formula**

   ```
   celsius = 5.0 / 9.0 * ( fahrenheit - 32 );
   ```

   class TestConvert:

   It has a *main()* **method to:**

   Create objects of classes *Celsius* **and** *Fahrenheit, where the values needed to initialize these objects are collected through dialog boxes*

   Executes the convertTo methods and the output is displayed in dialog boxes properly formatted.

2. **Generate the UML project with the UML plugin in IntelliJ (for reference view https://www.youtube.com/watch?v=ddHXKWguxWk) .Note, the UML plugin is readily available in IntelliJ Ultimate 2018**

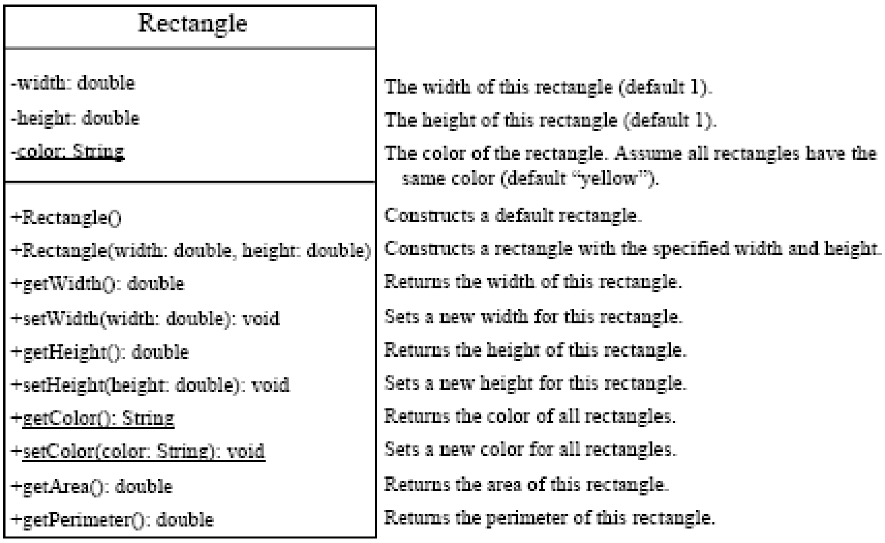
3. **Submit the Java application project and the UML project**

Problem 2.

Generate UML diagram for the following **class Rectangle** with the UML plugin in IntelliJ (for reference view **https://www.youtube.com/watch?v=ddHXKWguxWk**) .Note, the UML plugin is readily available in IntelliJ Ultimate 2018

Update the source code generated by the UML plugin to test the **Stock** class. Write a **StockTest** class to test the design and code. Collect the user input through dialog boxes (JavaFX)

Submit the Java application project and the UML project



Rectangle	
-width: double	The width of this rectangle (default 1).
-height: double	The height of this rectangle (default 1).
-color: String	The color of the rectangle. Assume all rectangles have the same color (default "yellow").
+Rectangle()	Constructs a default rectangle.
+Rectangle(width: double, height: double)	Constructs a rectangle with the specified width and height.
+getWidth(): double	Returns the width of this rectangle.
+setWidth(width: double): void	Sets a new width for this rectangle.
+getHeight(): double	Returns the height of this rectangle.
+setHeight(height: double): void	Sets a new height for this rectangle.
+getColor(): String	Returns the color of all rectangles.
+setColor(color: String): void	Sets a new color for all rectangles.
+getArea(): double	Returns the area of this rectangle.
+getPerimeter(): double	Returns the perimeter of this rectangle.

Problem 3.

Да се пресметне вероятността произволно избрано петцифрено число със следните свойства

- Първата цифра да е в интервала [3, 9]
- Втората цифра да е в интервала [2, 8]
- Третата цифра да е в интервала [4, 9]
- Четвъртата цифра да е в интервала [1, 6]
- Петата цифра да е в интервала [6, 9]

да се дели на 12. Заедно с пресметнатата вероятност да се изведе броят на числата с тези свойства, както и броят на числата със зададеното свойство.

Забележка: ***Използвайте*** ***`String.format()`***, подходящи **форматиращи спесификатори и специални (escape) символи**

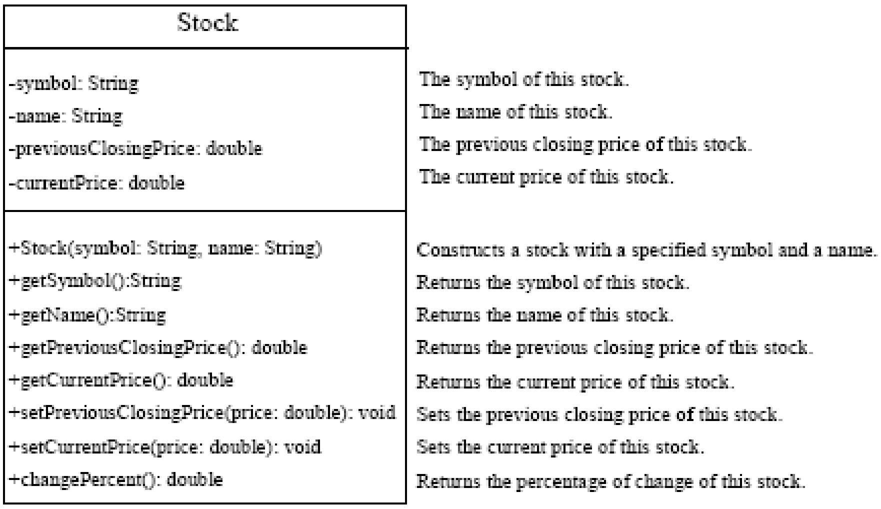
Create the Java application project for the following

Problem 4.

Generate UML diagram for the following **`class Stock`** with the UML plugin in IntelliJ (for reference view **`https://www.youtube.com/watch?v=ddHXKWguxWk`**) .Note, the UML plugin is readily available in IntelliJ Ultimate 2018

Update the source code generated by the UML plugin to test the **`Stock`** class. Write a **`StockTest`** class to test the design and code. Collect the user input through dialog boxes (JavaFX)

Submit the Java application project and the UML project



Stock	
-symbol: String	The symbol of this stock.
-name: String	The name of this stock.
-previousClosingPrice: double	The previous closing price of this stock.
-currentPrice: double	The current price of this stock.
+Stock(symbol: String, name: String)	Constructs a stock with a specified symbol and a name.
+getSymbol():String	Returns the symbol of this stock.
+getName():String	Returns the name of this stock.
+getPreviousClosingPrice(): double	Returns the previous closing price of this stock.
+getCurrentPrice(): double	Returns the current price of this stock.
+setPreviousClosingPrice(price: double): void	Sets the previous closing price of this stock.
+setCurrentPrice(price: double): void	Sets the current price of this stock.
+changePercent(): double	Returns the percentage of change of this stock.