

Sofia University
Department of Mathematics and Informatics

Course : [Applied OO Programming part 1](#)

Date: March 2, 2025

Student Name:

Lab No. 3

Problem No 1

Write a JavaFX application *class Geometry* to **draw a circle** in a *Group*. The *Circle* should be in the center of the *Scene* and have a *radius* which is **1/3 of the minimum** of the height and width of the *Scene*. Draw also a *line AB* **orthogonal** to the x- axis starting from a given x coordinate value. **Find the *intersection* points** of the line AB with the circle. In case there are no intersection points display an *Alert* dialog window with a warning message and quit the application. **Draw small circles** with centers at the intersection points **and display the text of the coordinates on the intersection**, appropriately formatted with 2 digits after the decimal point. Draw the circle and the line *AB* in different colors (red and blue). The *x- and y-coordinates of the line AB* should be **read before drawing the line with *TextInputDialog* and be validated to be in the interval $[0, w]$, where w is the width of the *Scene*.**

Problem No 2

In many applications we use a menu to allow the user select a set of options. Assume you have to write an application offering a set of choices to execute a program unit. In particular, there is an option allowing the user to update a data source represented by a floating point number. Additionally, you offer choices for the user to process this data source, for instance, compute $\sin()$, $\exp()$ or $\text{floor}()$ of the floating point number.

The user is enabled to execute these tasks by browsing a menu with options, displayed in Standard output. Let the menu has the following content:

1. *Update the floating point number*
2. *Compute and display $\exp(x)$*
3. *Compute and display $\sin(x)$*
4. *Compute and display $\text{floor}(x)$*
5. *Exit*

1. **Design an algorithm** and visualize it in a **UML activity diagram** [using draw.io](https://draw.io) to **complete the following tasks:**

- Displays the above menu until Option 5 is selected,
- Allows the user to input the desired option number
- Clears the standard output and displays the output of the executed option
- Clears the standard output before displaying the menu after the user has completed an option execution

- Employ the divide and conquer strategy to solve this problem
2. Write a **Java application**, comprising two classes, following the pattern represented in Lecture 2

The passive class, say `class ComputeByMenu` must have:

- Data member `double x`
- **A general purpose constructor**, allowing to initialize `x` `class ComputeByMenu`
- method `public void displayMenu() {}`, allowing to clear the standard output (output 50-60 empty lines) and **display the menu using Text block in Java**
- method `public void doSelection(int choice) {}`, executes the selected option `choice` making use of the current value of `x` and the methods of `class Math`
- method `public int getUserChoice() {}`, calls `displayMenu`, reads the user choice and calls `doSelection` to execute the selected option

The active class, say `class MenuTest` executes the following tasks:

- creates an instance of `class ComputeByMenu` with initial value for `x` equal to 0.00
- executes the method `getUserChoice()` of the thus created instance of `class ComputeByMenu`

Задача 3а

Напишете програма, която пресмята приближена стойност за константата e чрез представянето ѝ в следния безкраен ред **Дефинирайте алгоритъма** по който ще извършите пресмятането на константата e и го **визуализирайте като UML диаграма за дейност**

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots$$

Задача 3б

Пресметнете числото π от безкрайния ред

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \dots$$

Отпечатайте таблица, която показва приблизителната стойност на π получена при пресмятане до първия, втория,... n -тия член на този ред. **Прекратете табулацията** при получаване на стойност за π по-голяма от 3.14159. Отпечатайте **в края на табулацията най-малкия брой членовете** от този ред необходими за получаване за всяка от следните π стойности 3.14? 3.141? 3.1415? 3.14159?

1. **Дефинирайте алгоритъма** по който ще извършите табулацията и го **визуализирайте като UML диаграма за действие**
2. **Напишете програма** която реализира алторитъма в един клас (в `public static void main() {}` метода както в лекции 5)

[**Упътване:** Намерете формулата, по която се получава всеки пореден член на безкрайния ред в зависимост от поредния му номер. Нечетните числа се представят като $2n + 1$ за $n = 0, 1, 2, \dots$]

Как да се промени логиката на изпълнение на задачата, за да се гарантира, зададена от потребителя точност $0 < \epsilon < 1$ на пресмятанията (брой цифри след десетичната запетая, съвпадащи с точното решение)? Сравнете полученият резултат с `Math.PI`.

Задача 3с

Напишете програма която пресмята стойността на **sin(x)** и **cos(x)** чрез представянето ѝ в следния безкраен ред. **Дефинирайте алгоритъма** по който ще извършите пресмятането на **sin(x)** и **cos(x)** и го **визуализирайте** като **UML** диаграма за дейност

$$\sin x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots \quad \text{for all } x$$

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \quad \text{for all } x$$

Упътване: Прекратете пресмятане на членове от безкрайния ред, когато разликата в абсолютната стойност на два съседни члена в редицата стане по-малък от 0.001

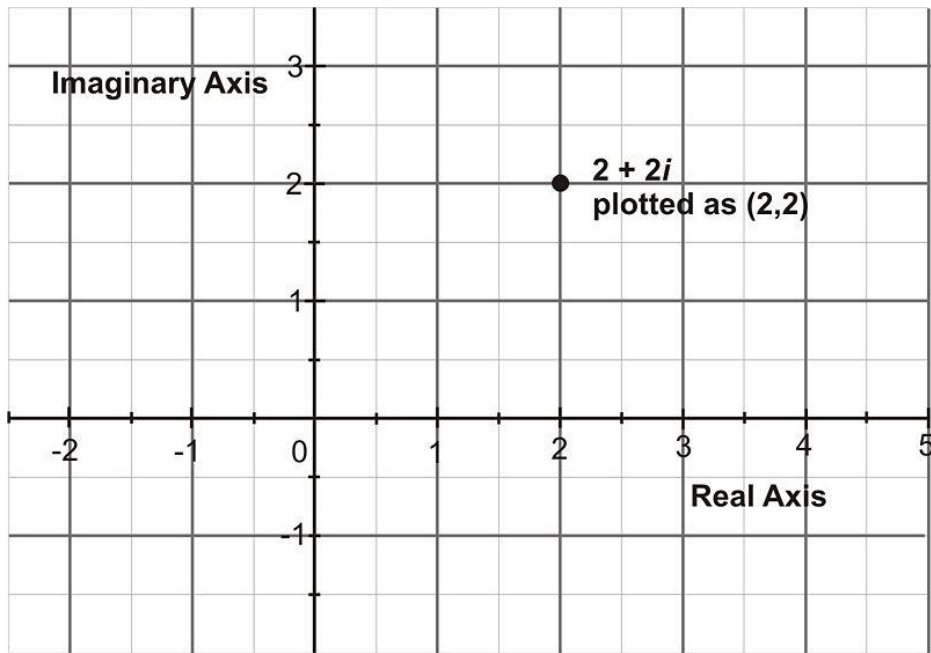
Problem No. 4

Submit both the Java application project for the following class `ComplexNumber`

ComplexNumber
imaginaryPart:double realPart:double
ComplexNumber(double,double) add(ComplexNumber):ComplexNumber getAngle():double getImaginaryPart():double getMagnitude():double getRealPart():double

To compute the angle in the trigonometric representation of a complex number use the static method `Math.atan2(double x, double y)` explained as

https://www.tutorialspoint.com/java/lang/math_atan2.htm



If you are given r and θ then use $x = r \cos \theta$ and $y = r \sin \theta$ and the complex number $z = x + i y$ is represented in trigonometric form as $z = r(\cos \theta + i \sin \theta)$, where $r = \sqrt{x^2 + y^2}$ and $\tan \theta = \left| \frac{y}{x} \right|$

Write a `ComplexNumberTest` class to test the design and code. Collect the user input through dialog boxes (JavaFX)

Problem No. 5

Write a Java program that displays a calendar for a month. The program reads the number of the month and the year:

Mon	Tue	Wed	Thr	Fri	Sat	Sun
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Calendar for MARCH 2018						
MON	TUE	WED	THR	FRI	SAT	SUN
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

Problem No. 6

Write a static method `printListOfAllFriday13 ()` in a Java class to create a list of all the Fridays in a given year that fall on the 13th day of a month. Test the method in the `main()` by generating the year as a random number with class `Random()` in the interval [1900- 2020].