

**Sofia University**  
**Department of Mathematics and Informatics**

**Course : OO Programming with Java**

**Date: October 8, 2018**

**Student Name:**

**Lab No. 2b**

**Задача 1**

Да се табулират стойностите на функцията  $f(x) = \frac{|x-2|^2}{x^2+1}$

по зададени **начална** и **крайна** стойност за интервала от стойности на  $x$  (числа в плаваща стойност с двойна точност), както и **брой** стъпки за дискретизация на интервала от стойности на  $x$ .

- За всяка стойност на  $x$  да се извежда и съответната стойност на функцията  $f(x)$  за това  $x$  - използвайте специален символ за табулация
- Да се използват подходящи флагове за форматиране на тези стойности с **4 знака след десетичната запетая**.
- Дискретизираните стойности **да се извеждат в серии от по 20**. В края на всяка серия да се отпечата съобщение *"Press return to continue ..."* и да се изчака потребителят да въведе *Return* преди да се покаже **следващата серия** от стойности

1. **Да се напише алгоритъм** за изпълнение на така изискваната табулация и да се визуализира като **UML диаграма** за действие ( на хартия)

2. Задачата **да се структурира** като програма на **Java**, състояща се от два класа **по аналогия с модела**, използван на **Лекции**. Нека *class Table* да реализира табулацията. Нека **този клас да има**

- за **клас данни**- три променливи за **началната и крайната** стойност за интервала от стойности на  $x$ , както и за брой **стъпки** за дискретизация
- **конструктор за общо ползване**, който инициализира тези данни
- метод *public void makeTable() {}*, който да извършва табулацията, съответстваща на стойностите на клас данните по създадения алгоритъм

Другият клас (активният клас или т. нар клас за тестване, например, именувайте го *class TableTest*) да реализира

- **прочитане** на **начална** и **крайна** стойност за интервала от стойности на  $x$ , както и брой **стъпки** за дискретизация с диалогов прозорец (*JavaFX*)
- **да се провери дали въведената начална стойност е по- малка от крайната и в противен случай да се разменят**
- **да създаде обект** от пасивния клас (*class Table*) като **използва конструктора му за общо ползване** за да **инициализира този обект** с така въведените стойности за краища на интервала и стъпката

- изпълнява метода `makeTable()` на така създадения обект за `class Table`

## Задача 2

Напишете **Java приложение** за пресмятане на  $\sin(x)$  чрез безкрайния ред:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

с точност  $0 < \varepsilon < 1$  задавана от потребителя т.е. **точността се счита постигната**, когато се включат толкова члена на безкрайния ред при които абсолютната стойност на разликата за две последователни частични суми става е по- малка от  $\varepsilon$

**Да се напише алгоритъм** за изпълнение на така изискваното пресмятане и да се визуализира като **UML диаграма** за действие (като **UML** ( на хартия))

Нека  $\sin(x)$  се пресмята в отделен `class ComputeSin`, който има `double` данна `x`, `set` и `get` методи за `x` и метод `double computeSin()`, който пресмята  $\sin(x)$  по горната формула.

Напишете `class ComputeSinTest`, който табулира стойностите на  $\sin(x)$  в интервала  $[0, \pi/2]$  **форматиране с 3 цифри след десетичната запетая** като използвате стъпка за `x=0.1`

## Задача 3

Едно цяло **число** се нарича **перфектно**, ако се **представя като сума на множителите си**. Например, 6 е перфектно число, понеже,

$$6 = 1 + 2 + 3.$$

- Напишете метод
 

```
public boolean isPerfect( int number)
```

 който определя дали зададения аргумент `number` е **перфектно число** или не е перфектно число като съответно връща `true` или `false`.
- Нека са дефинирани **константи** `NUMBER_PER_LINE = 5` и `NUMBER_OF_PERFECTS= 10000`
- Напишете още метод
 

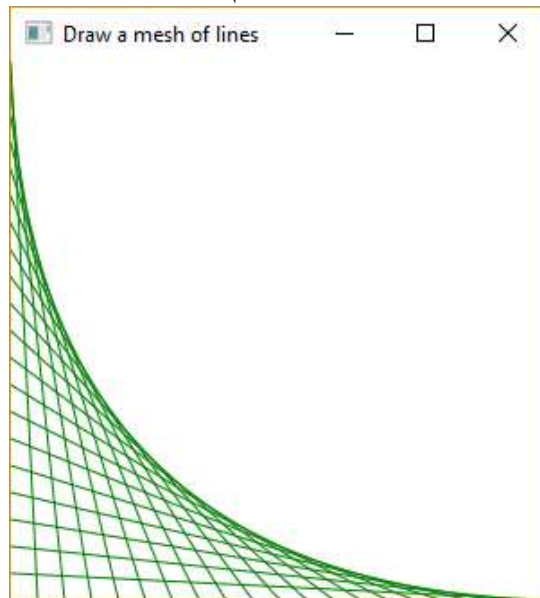
```
public void showMultipliers( int number)
```

 който **извежда табулирано в редове** от по `NUMBER_PER_LINE` числа **множителите** на зададения аргумент `number`.
- Напишете **алгоритмите** и тяхната **UML визуализация** за тези два метода
- Използвайте тези методи в **Java приложение**, което **определя** и **извежда** всички перфектни числа между 1 и `NUMBER_OF_PERFECTS`.. За всяко

*перфектно* число открито с `isPerfect()` изведете за проверка и съответните му множители, чрез `showMultipliers()`

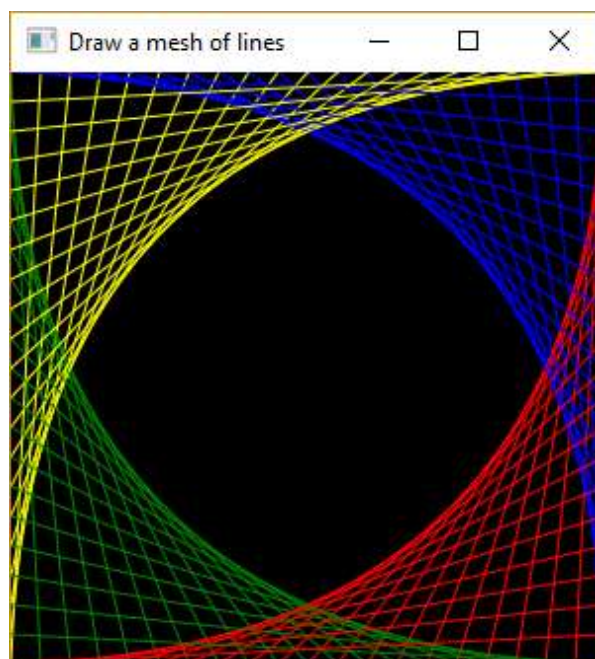
#### Задача 4a

Да се напише JavaFX приложение, което реализира следната графика. Да се използва изчертаване на линии в цикъл



#### Задача 4b

Да се напише JavaFX приложение, което реализира следната графика. Да се използва изчертаване на линии в цикъл



### **Задача 5**

**Аналогично на Задача 4 да се напише JavaFX приложение, което реализира следната графика. Да се използва изчертаване на линии в цикъл**

