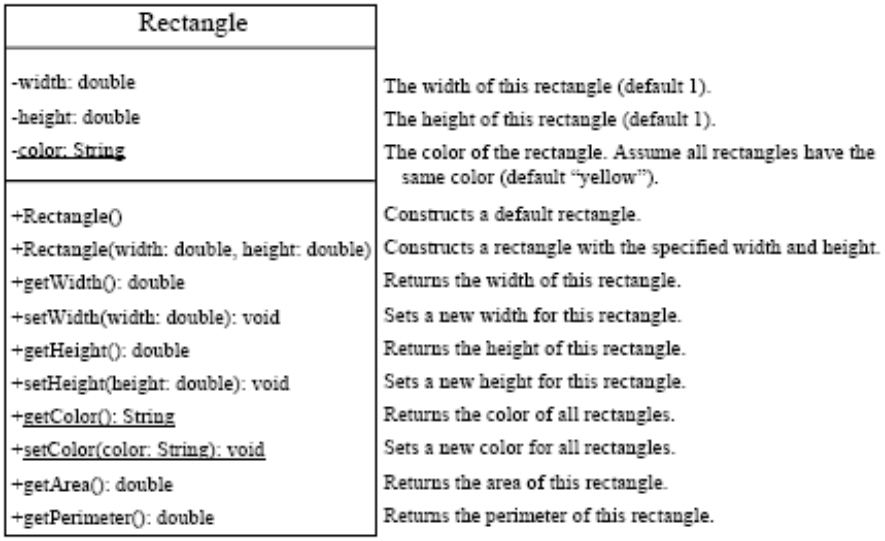
Sofia University
Department of Mathematics and Informatics

Course : Applied OO Programming part 1
Date: February 24, 2025
Student Name:

Lab No. 2

Задача 1a

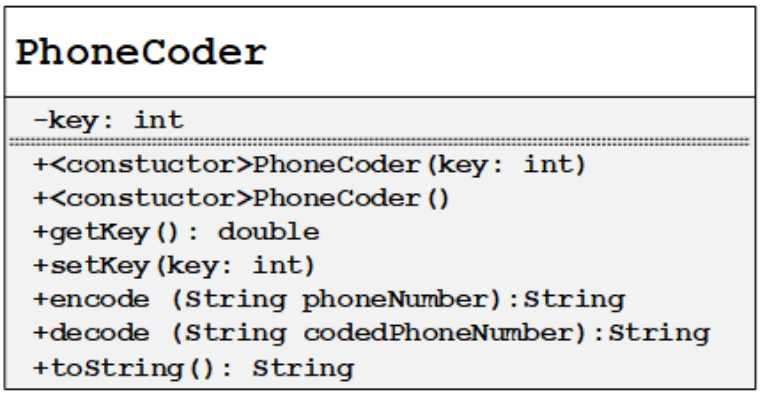
Създайте UML клас диаграма на `class Rectangle` на IntelliJ като възпроизведете следния обектно ориентиран модел на този клас.



Напишете приложение на Java на IntelliJ, което съдържа клас `Rectangle` в пакета `shape`
Напишете `class RectangleTest` за да тествате приложението с данни, въвеждане от стандартен вход.

Задача 1b

Една компания иска да предава данни по телефон, но се опасява, че телефоните й се подслушват. Да предположим, че данните за телефонни номера се представят като четирицифрени цели десетични числа и компанията ви възлага да се напишете приложение на Java, което кодира тези данни, така че те могат да бъдат защитени при преноса им по телефонните линии.
Вашата програма трябва реализира клас **`PhoneCoder`** със следната UML клас диаграма



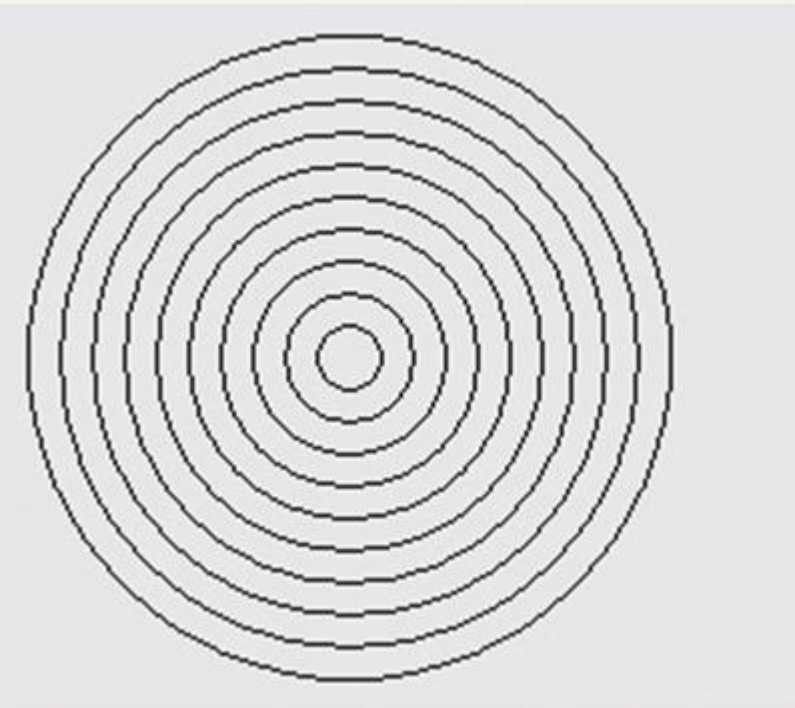
- Класът **`PhoneCoder`** има гет- и сет-свойство **`key`**, конструктори и **`toString()`** метод, който връща форматиран стринг от вида "`PhoneCoder : <key>`", където `<key>` е текущата стойност данната **`key`**. Данната **`key`** е ключът за кодиране на телефонните номера и представлява цяло число в интервала `[1, 9]`.
- Методът **`encode()`** взима за параметър цяло число, образувано от четирицифрения телефонен номер и връща четирицифрено цяло число, което е кодирания телефонен номер. Кодирането се извършва като се извличат четирите цифри на зададения телефонен номер. Всяка от тези цифри се заменя със стойността модул по 10 от на сумата от стойността на цифрата и ключа **`key`**. Така, ако имате цифра 6, то тя се заменя със стойността на `( 6 + key ) mod 10`. Допълнително, първата и третата се разменят, също така втората и четвъртата се разменят и така полученото цяло четирицифрено число представлява кодирания телефонен номер.
- Методът **`decode()`** взима за параметър цяло число, образувано от кодиран четирицифрен телефонен номер и връща четирицифрено цяло число ,което е декодирания телефонен номер. Декодирането се извършва като се извличат четирите цифри на кодирания телефонен номер. Всяка от тези цифри се заменя със стойността на разликата на цифрата и ключа **`key`**. Към тази разлика се добавя **`10`**, ако цифрата е по- малка от ключа **`key`**. Така, ако имате цифра **`6`** в кодирания телефонен номер, то тя се заменя със стойността на `( 6 - key )`, когато **`6`** е по-голямо или равно на **`key`** . Допълнително, третата и първата цифри се разменят, също така четвъртата и втората цифри също се разменят и така полученото цяло четирицифрено число представлява декодирания телефонен номер.

Напишете клас **`PhoneCoderTest`** където да тествате методите на клас **`PhoneCoder`**. Допълнително, обмислете необходимите промени в случаите, когато трябва да се кодира телефонен номер, който започва с нули.

Задача 2

Променете програмата ***`DrawLines`*** от ***`Lecture2b`*** да може да рисува **12 концентрични окръжности (респ. квадрати)** с център- средата на графичния прозорец , както е показано на **фигурата по- долу.** Най- вътрешната окръжност**(респ.**

вътрешен квадрат) да има **диаметър (страна) от 20 пиксела**, а всяка **следваща окръжност (квадрат) да има с 20 пиксела по- голям диаметър (страна)**. Започнете програмата си с **намиране на центъра** на своята сцена. **Използвайте цикъл** управляван от брояч за решение на задачата.



Нарисувайте също вертикалния и хоризонталния диаметър през центъра на външната окръжност **с червен цвят.**

Задача 3

Напишете програма която пресмята стойността на $\mathbf{e^x}$ чрез представянето ѝ в следния безкраен ред. **Дефинирайте алгоритъма** по който ще извършите табулация**.** **Дефинирайте алгоритъма** по който ще извършите пресмятането на $\mathbf{e^x}$ и го **визуализирайте като UML диаграма за дейност**

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

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

Задача 4

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 управляващи (escape) символ**

Задача 5a

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**

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

Задача 5b

Пресметнете числото π π от безкрайния ред

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

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 стойностти 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,`**…]

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

Задача 5c

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**

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

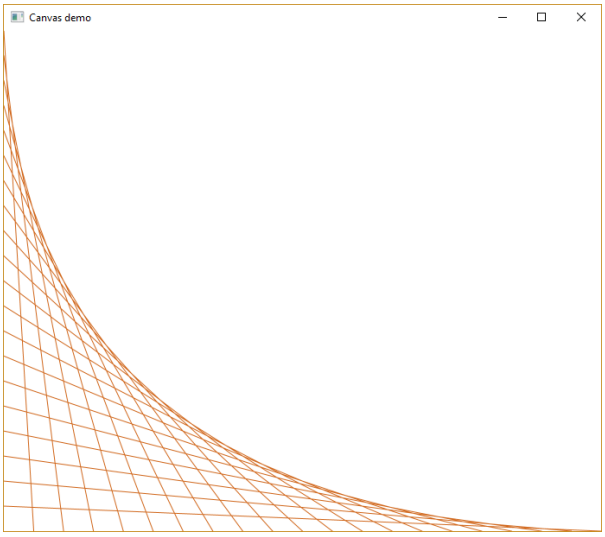
$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \qquad \text{for all } x$$

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

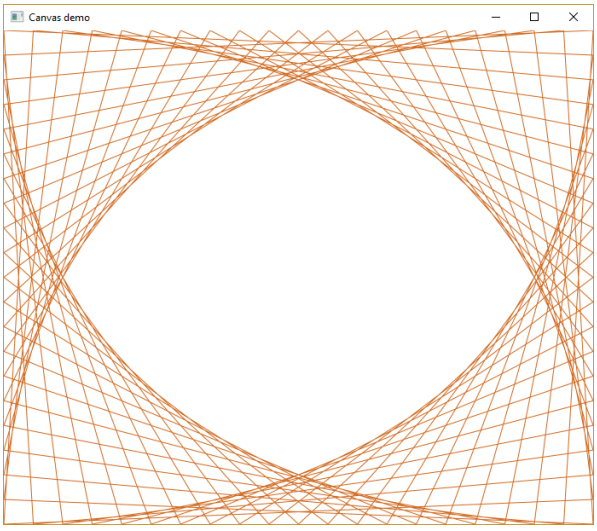
Задача 6

Напишете JavaFX приложение подобно на примера **DrawLines** в Lecture2b, което пресъздава следните геометрични фигури като използвате ***for*** **цикли за повторение**

a)

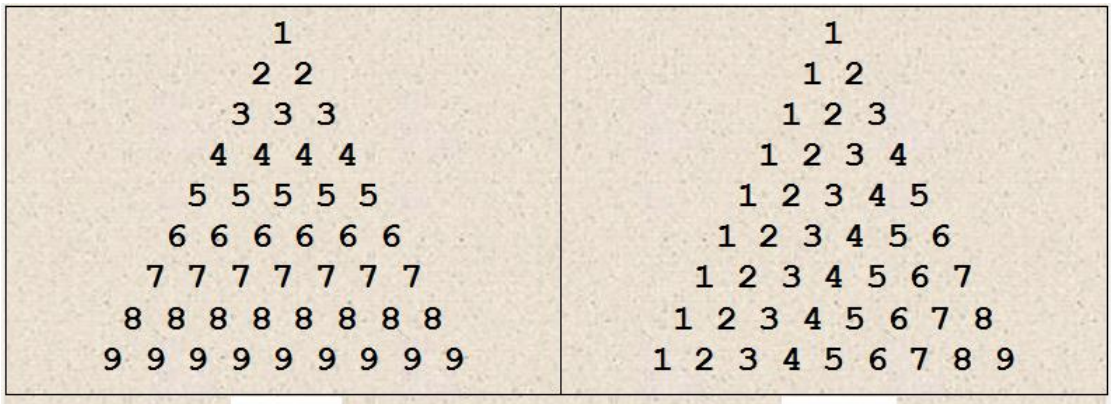


b)



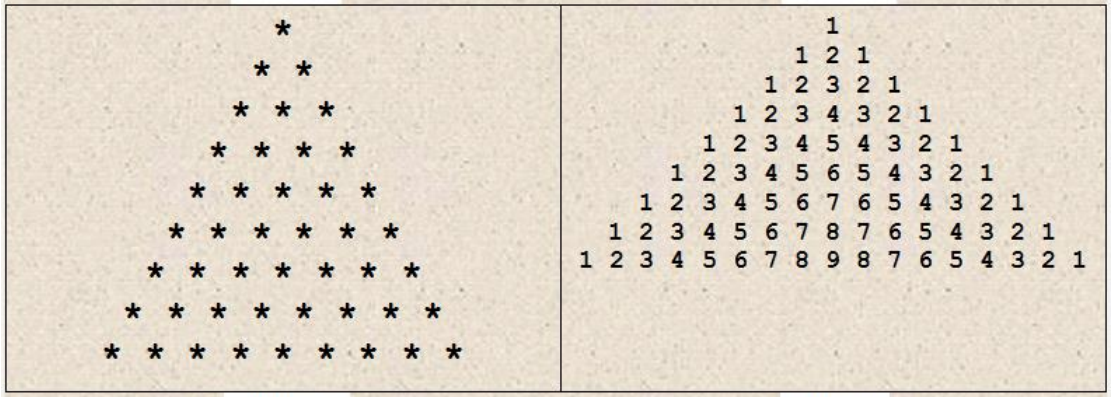
Задача 7

Напишете Java приложение, което пресъздава следните пирамиди от числа и символи като използвате ***for*** **цикли за повторение**



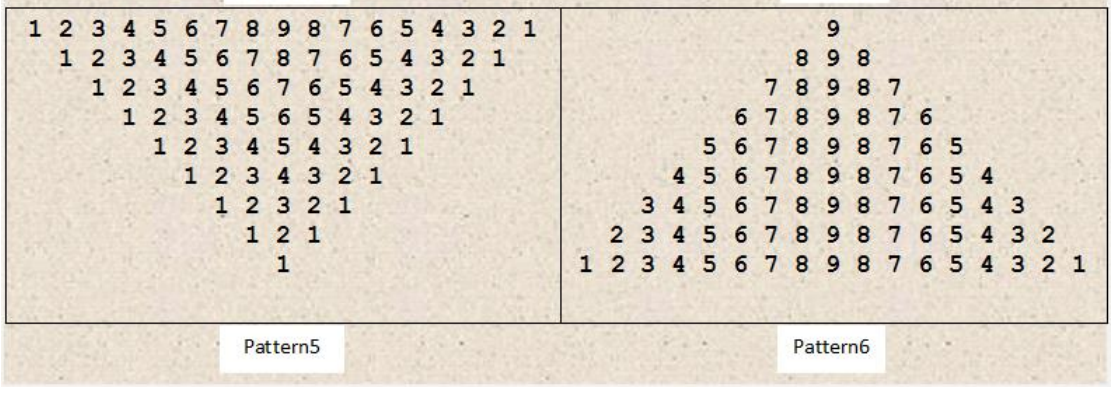
Pattern1

Pattern2



Pattern3

Pattern4



Pattern5

Pattern6