

Домашна работа #7

Краен срок: 17.12.2018

Важно:

1. Всяко решение трябва да бъде в **отделен NetBeans проект**.
2. Името на пакета задължително трябва да е **bg.unwe**
3. **Имената на класовете** трябва да съвпадат точно с посоченото във всяка задача.
4. За тези задачи се очаква да напишете кода си **изцяло в main метода**.
5. Навсякъде, където е посочено че трябва да се изпише някакъв резултат, е необходимо **текстовете да съвпадат** напълно.

Задача 1 (Homework 7-1)

Име на класа: **SumIfLarger**

Напишете програма, в която потребителят въвежда поредица от **положителни цели числа**, които се сумират. Сумирането **да спира** когато потребителя въведе число, което е по-малко от предходното.

Например, ако потребителят въведе 1, 2, 3, 4, 2, то последната 2-ка е по-малка от 4 и цикълът следва да прекъсне и да се визуализира сумата само на числата до този момент, т.е. $1+2+3+4$.

За да може да сравнявате с предходното число, е необходимо преди цикъла да си създадете допълнителна променлива, в която да го съхранявате.

```
run:
Next: 1
Next: 2
Next: 3
Next: 4
Next: 2
Result: 10
BUILD SUCCESSFUL (total time: 8 seconds)
```

Примерен алгоритъм за решение:

```
Декларираме променлива за сумата = 0;
```

Декларираме променлива за предходното число = 0;

Декларираме променлива за текущото число = 0;

Докато (текущото число \geq предходното число):

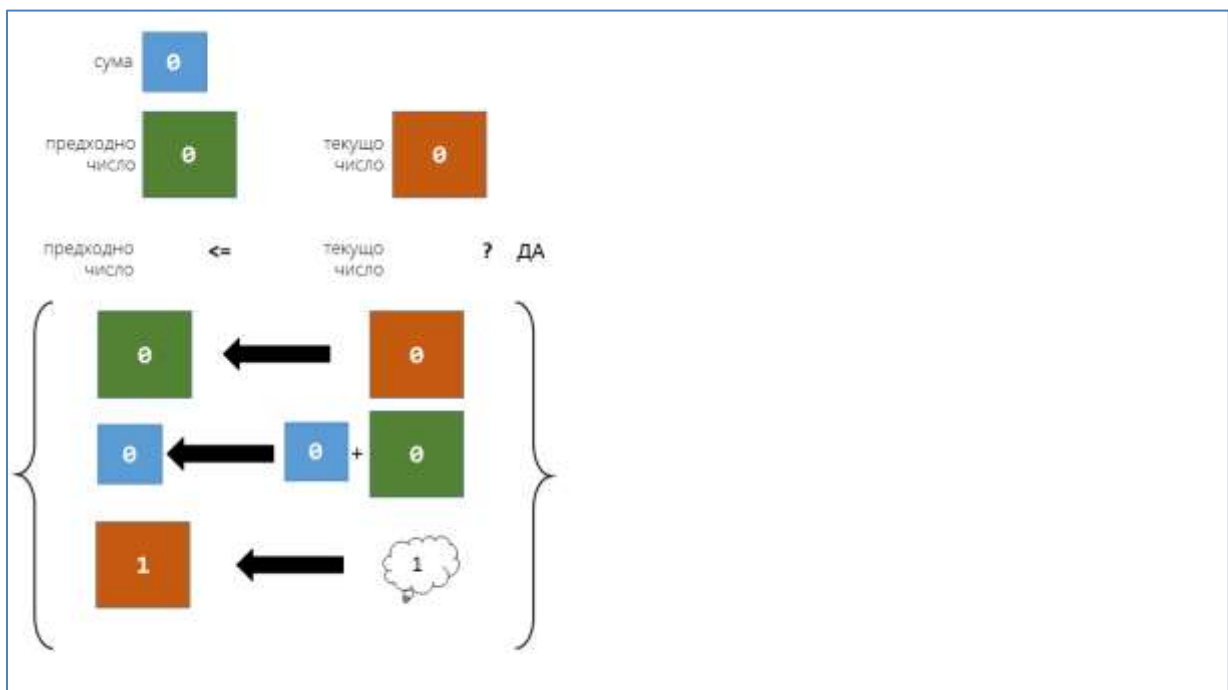
Присвояваме стойността на текущото число на предходното число;

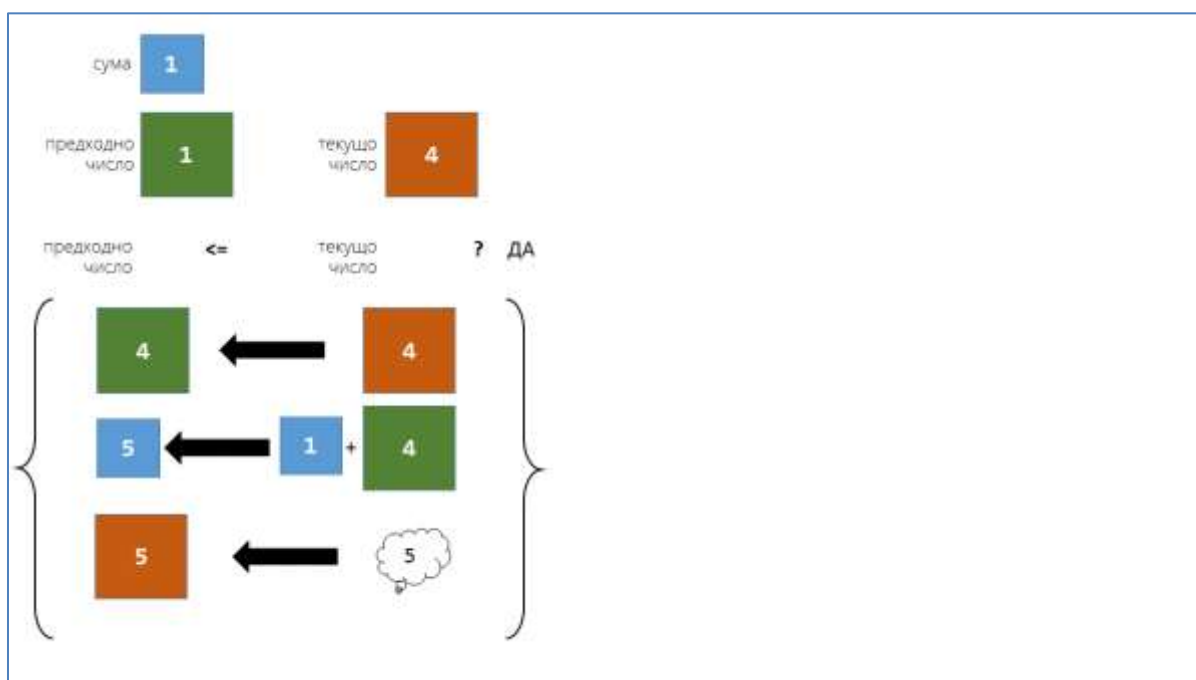
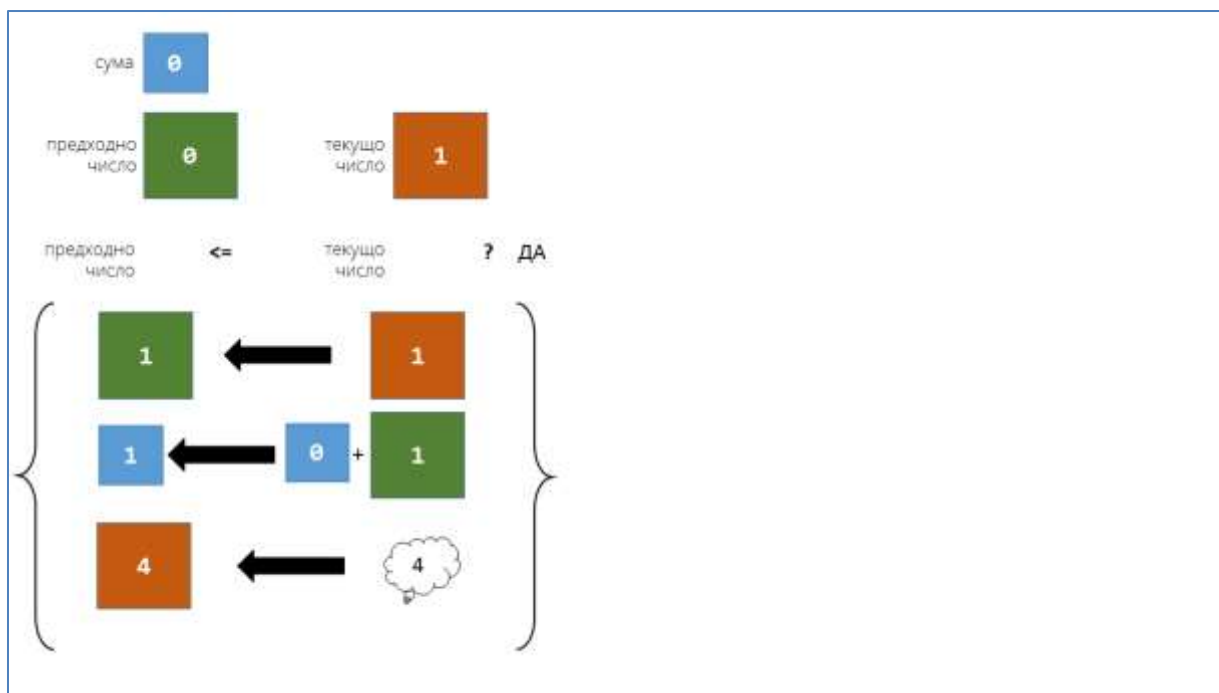
Прибавяме предходното число към общата сума;

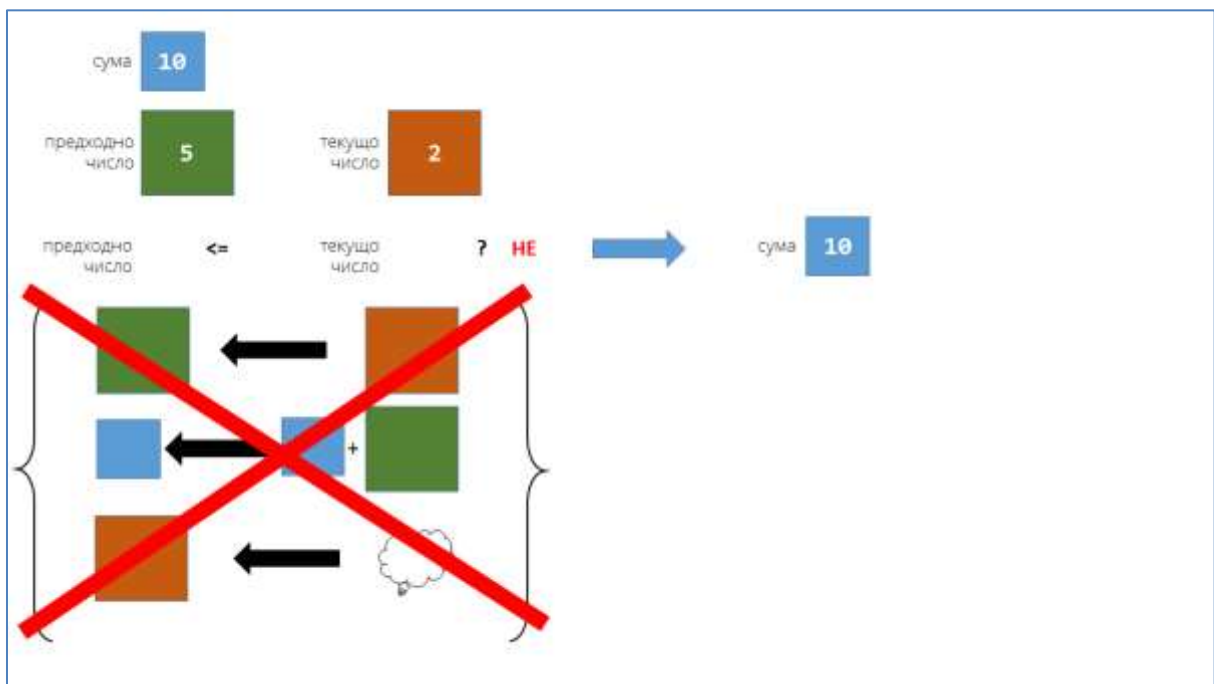
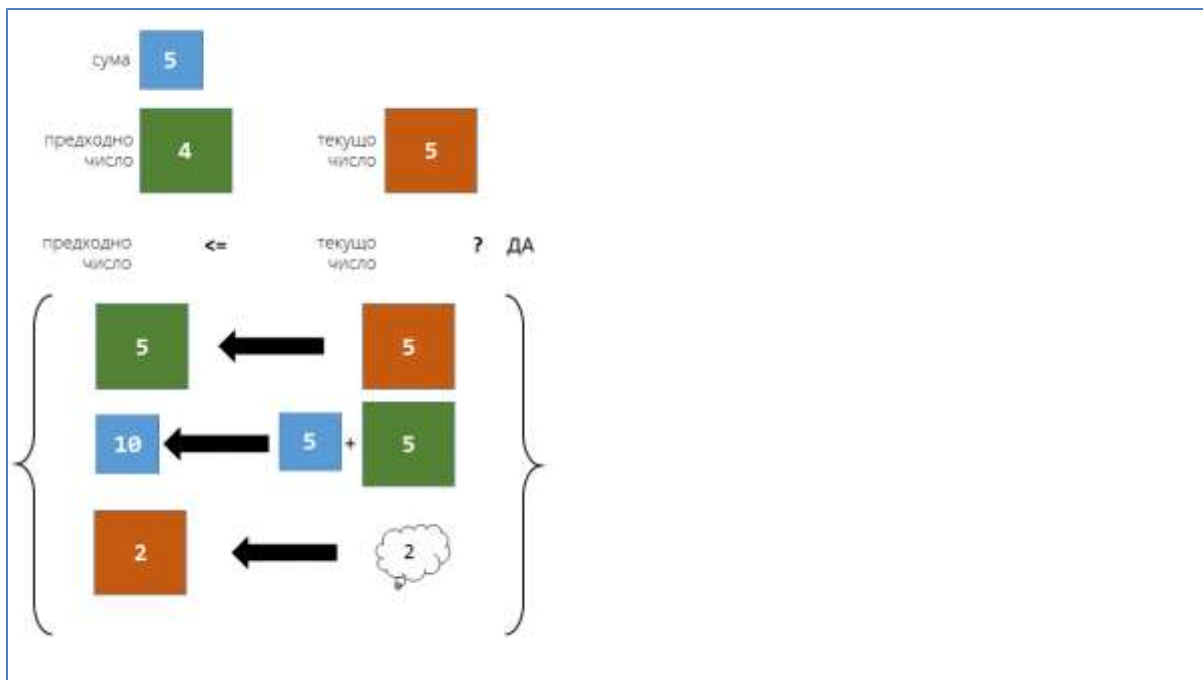
Питаме за въвеждане на число и го съхраняваме като текущо число;

Когато сме приключили, показваме сумата.

Ако въведем 1, 4, 5, 2, последователността би била следната:







Задача 2 (Homework 7-2)

Име на класа: **Fibonacci**

Напишете програма, която получава от потребителя число N и съответно извежда N на брой елемента от **поредината на Фибоначи**. При поредината на Фибоначи **всеки** следващ елемент е сумата от предходните два, например: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34,

55 и т.н. Т.е. при всяко завъртане на цикъла, за да получите текущата стойност трябва да сумирате предходните две числа.

Необходимо е преди цикъла да си създадете **две променливи, в които да съхранявате предходните две числа**. Може да ги инициализирате с 0 и 1, а след това трябва при всяко завъртане на цикъла да ги обновявате коректно.

```
run:
Limit: 10
0
1
1
2
3
5
8
13
21
34
BUILD SUCCESSFUL (total time: 2 seconds)
```

Примерен алгоритъм за решение:

Питаме колко елемента да изчислим;

Декларираме променлива за първото число от поредицата = 0;

Декларираме променлива за второто число от поредицата = 1;

Показваме на екран първото и второто число;

Повтаряме X пъти следното (където X е желаният брой елементи минус 2, защото вече показахме два елемента):

 Декларираме си текущо число като сума от първото и второто число;

 Показваме текущото число;

 Присвояваме второто число на първото число;

 Присвояваме текущото число на второто число;

Задача 3 (Homework 7-3)

Име на класа: **ConeVolumeLoop**

Напишете програма, която изчислява обема на конус за всички стойности на радиус между 1 и N включително и всички стойности на височина между 1 и M включително.

Стойностите на N и M следва да се въвеждат от клавиатурата.

Може да използвате **вложен цикъл**, за да получите комбинацията от различни радиуси и височини.

Примерен вид на изхода на програмата при въведени максимален радиус 3 и максимална височина 5:

```
run:
Max Radius: 3
Max Height: 5
radius 1, height 1:
volume: 1.05
radius 1, height 2:
volume: 2.09
radius 1, height 3:
volume: 3.14
radius 1, height 4:
volume: 4.19
radius 1, height 5:
volume: 5.24
radius 2, height 1:
volume: 4.19
radius 2, height 2:
volume: 8.38
radius 2, height 3:
volume: 12.57
radius 2, height 4:
volume: 16.76
radius 2, height 5:
volume: 20.94
radius 3, height 1:
volume: 9.42
radius 3, height 2:
volume: 18.85
radius 3, height 3:
volume: 28.27
radius 3, height 4:
volume: 37.70
radius 3, height 5:
volume: 47.12
BUILD SUCCESSFUL (total time: 4 seconds)
```

Задача 4 (Homework 7-4)

Име на класа: **QuadraticEquationsLoop**

Разполагаме с код на програма, която решава квадратни уравнения за комбинация от стойности на трите променливи a , b и c между 1 и N (вкл.), където N се въвежда от клавиатурата. Копирайте кода и поправете грешките, така че да се получи коректен резултат, какъвто е показан по-долу.

```
package bg.unwe;

import java.util.Scanner;

public class QuadraticEquationsLoop {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        int limit;
        Scanner s = new Scanner(System.in);
        System.out.print("Limit: ");
        limit = s.next();

        for (int a = 0; a < limit; a++);
        for (int b = 0; b < limit; b++) {
            for (int c = 0; c < limit; c++) {
                System.out.printf("a = %f, b = %f, c = %f\n", a,
b, c);
            }
            double d = b * b - 4 * a * c;
            if (d < 0) {
                System.out.println("No solution");
            } else if (d == 0) {
                double x1 = -b / (2 * a);
                System.out.println("x1 = " + x1);
            } else {
                double x1 = (-b - Math.sqrt(d)) / (2 * a);
                double x2 = (-b + Math.sqrt(d)) / (2 * a);
                System.out.println("x1 = " + x1);
                System.out.println("x2 = " + x2);
            }
        }
    }
}
```

Примерен вид на изпълнението на програмата при въведена горна граница 3:

```
Limit: 3
a = 1, b = 1, c = 1
No solution
a = 1, b = 1, c = 2
No solution
a = 1, b = 1, c = 3
No solution
a = 1, b = 2, c = 1
x1 = -1.0
a = 1, b = 2, c = 2
No solution
a = 1, b = 2, c = 3
No solution
a = 1, b = 3, c = 1
x1 = -2.618033988749895
x2 = -0.3819660112501051
a = 1, b = 3, c = 2
x1 = -2.0
x2 = -1.0
a = 1, b = 3, c = 3
No solution
a = 2, b = 1, c = 1
No solution
a = 2, b = 1, c = 2
No solution
a = 2, b = 1, c = 3
No solution
a = 2, b = 2, c = 1
No solution
a = 2, b = 2, c = 2
No solution
a = 2, b = 2, c = 3
No solution
a = 2, b = 3, c = 1
x1 = -1.0
x2 = -0.5
a = 2, b = 3, c = 2
No solution
a = 2, b = 3, c = 3
No solution
a = 3, b = 1, c = 1
No solution
a = 3, b = 1, c = 2
No solution
a = 3, b = 1, c = 3
No solution
a = 3, b = 2, c = 1
No solution
a = 3, b = 2, c = 2
No solution
a = 3, b = 2, c = 3
No solution
a = 3, b = 3, c = 1
No solution
a = 3, b = 3, c = 2
No solution
a = 3, b = 3, c = 3
No solution
```