VENTSPILS AUGSTSKOLA in collaboration with accenture

Seminar 1

Introduction to JAVA programming

Open Eclipse, create new project “Seminar_1” with package name javasem01 and class JavaSem01.

Exercise 0 - warmup

Output marathon names with the corresponding times in two columns where the output form of each row is given as follows [*Name, <time>*]:

```
String[] names = { "Elena", "Thomas", "Hamilton", "Suzie", "Phil", "Matt",
"Alex", "Emma", "John", "James", "Jane", "Emily", "Daniel", "Neda", "Aaron",
"Kate" };

int[] times = { 341, 273, 278, 329, 445, 402, 388, 275, 243, 334, 412, 393,
299, 343, 317, 265 };
```

Exercise 1

```
//position calculator in case of a falling point-like object
//This is an example with a partial implementation in main method
class Seminar_1_PART2 {
   public static void main(String[] arguments) {
      double gravity = -9.81; // Earth's gravity in m/s^2
      double initialVelocity = 0.0;
      double fallingTime = 10.0;
      double initialPosition = 0.0;
      double finalPosition = 0.0;
      System.out.println("Pos in "+fallingTime+"sec :"+finalPosition);
   }
}
```

Compute the position after 10 seconds. Compute using following equation:

$x(t) = 0.5 \times a*t^2 + v_0*t + x_0$

Implementation:

```
double positionCalc(double gravity, double initialVelocity, double
initialPosition, double fallingTime)
```

Exercise 2

Compute factorial:

$f = N! = 1*2*3*...*N;$

- Using for loop
- Using recursive function

Implementation:

`int factorialForLoop(int N)` and `int factorialRecursive(int N)`

If input parameter is given negative, return 0;

Exercise 3

Working with arrays:

- Create function that:
  - Create double type array and fills it with **N** random double values in a range between **lower** and **upper**. (if lower bound is higher than the upper bound, return empty array);
  - Return the smallest & biggest values
  - Return Mean value
  - Implement Bubble sort (return sorted array in ascending order)

Implementation:

```
double[] generateArray(int N, double lower, double upper)
double getMean(double[] array)
double getMin(double[] array)
double getMax(double[] array)
double[] arraySort(double[] array)
```

Exercise 4

1. Create a multidimensional rectangular matrix (2 dimensional array) of size NxN

	A	B	C	D	E
0	10	12	43	11	22
1	20	45	56	1	33
2	30	67	32	14	44
3	40	12	87	14	55
4	50	86	66	13	66
5	60	53	44	12	11

2. Output the dot product of i-th row and j-th column

i=1...N

j=1...N

Implementation

```
double[][] generateMatrix(int N)
double getProduct(double[][] matrix, int i, int j)
```

Exercise 5

Part 1

Simulate N coin flips where each coin flip yield 0 (head) or 1 (tails) randomly. Count the number of zeros and ones. What is the ratio of heads and tails if N = 10, 100, 1000 and 10000?

Implementation

```
double[] coinFlip(int N)

The method returns an array of three elements where:
element at index 0 - number of heads
element at index 1 - number of tails
element at index 2 - ratio: nheads/ntails
```

Part 2

Write a program that simulates rolling dice, i.e. gives random number from 1 - 6. Simulate N rolls and count the number of each occurrence (histogram).

Implementation

```
int[] rollDice(int N)

where elements at indexes 0 - 5 returns the number of occurences
for each case (1 - 6)
```

Part 3

Roll two dices until both are 6 (12 together). Output the number of tries until the desired combination became true.

Implementation

```
int roll2Dices()
```

Exercise 6

Assume following byte array:
72, 101, 108, 108, 111, 33, 32, 77, 121, 32, 115, 107, 105, 108, 108, 115, 32, 97, 114, 101, 32, 103, 114, 101, 97, 116, 32, 97, 108, 114, 101, 97, 100, 121, 33

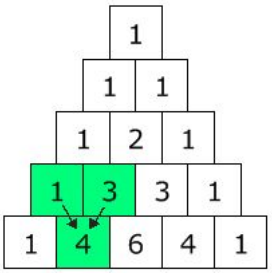
Output the corresponding `char` values as a string!

Implementation

```
String getTextFromBytes(byte[] array)
```

Exercise 7

Implement function that outputs the Pascal's Triangle:



Function takes N (level or number of rows) an input parameter. One might use previously implemented factorial methods.

Mathematically:

$$a_{n\,r} \equiv \frac{n!}{r!\,(n-r)!} \equiv \binom{n}{r},$$

where $\binom{n}{r}$ is binomial coefficient. n - index of the row, r - index in the element

Example of implementation:

```
int level = 6;
String pascalsTriangle(int level);
```

Function output:
[1, 6, 15, 20, 15, 6, 1]

Exercise 8*

Try to execute the expression written as a String:

```
String inputEquation = "1 - 3 * 18 / 4 + 2";

Output: -10.5
```

Implementation

```
double executeStringEquation(String inputEquation)
```

Exercise 9*

Write following program that is able to process following sets represented as `Strings`:

Input	Output
[1, 2, 3] + [3, 5, 7]	[1, 2, 3, 5, 7]
[10, 9, 8, 7] * [2, 4, 6, 8]	[8]
[5, 10, 15, 20] - [10, 20, 80]	[5, 15, 80]

Unification of sets: +
Intersection: *
Symmetric difference: -

Implementation

```
String setOperations(String input)
```