KAZAKH KBTU BRITISH TECHNICAL UNIVERSITY

Object-Oriented programming

Laboratory work #2

Problem #1

Implement a class `Student`. Student has a `name`, `id` and a `year of study`. Provide a constructor with two parameters and create methods to access name, id and increment the year of study.

Problem #2

Write a class `StarTriangle` that can be used to generate the following triangular shape :

[*]
[*][*]
[*][*][*]
[*][*][*][*]
....

- Your class should have a constructor with a parameter width, that indicates the number of [*] in the last row of the triangle.
- Moreover, there must be a method `toString()` that computes a string representing the triangle and returns a string consisting of [*] and newline characters.

Class usage:

```
StarTriangle  small = new StarTriangle(3);
System.out.println(small.toString());
```

The output is:

[*]
[*][*]
[*][*][*]

Problem #3

Write a class `Data` that computes information about a set of data values. It should have:

- 3 private fields: 2 of type double and 1 of type int (you need to guess what to store in these fields)

- Constructor that constructs an empty data set.
- Method that adds a value to a data set.
- Method that returns average of the added data or 0 if no data has been added.
- Method that returns the largest of the added data.

Note: You are not allowed to store ALL values ! Do not use arrays or something like this!

Then write a class `Analyzer` that uses `Data` class described above to compute the average and maximum of a set of input values.

- Use `Scanner` to take data from user
- The process of taking inputs from user must continue until the user types "Q" instead of a number.

For example:

Enter number (Q to quit): 10
Enter number (Q to quit): 0
Enter number (Q to quit): -1
Enter number (Q to quit): Q
Average = 3.0
Maximum = 10.0

Problem #4

Write a class `Time` that has the following fields : `hour, minute, second`. Also, you need to have a corresponding constructor and method that set the time according to provided hour, minute and second parameters (check for any invalid input). Moreover, you need to have two methods for time format conversion (Universal and Standard) and method to add two `Time` objects. It can be either static (in this case there will be 2 parameters of type `Time`, or it can be non-static instance method taking one `Time` parameter.

Universal: 14:23:22
Standard: 02:23:22 PM
Class usage:

```
Time t = new Time(23, 5,6);
System.out.println(t.toUniversal())// prints "23:05:06"
System.out.println(t.toStandard());//prints "11:05:06 PM"
Time t2 = new Time(4,24,33);
t.add(t2);
```

Problem #5

Write any class at you choice and show appropriate usage of: enums, "static" and "final" modifiers,"this" keyword (2 usages), read-only fields, methods overloading, initialization block.