May 12, 2022

Task 7

Create a class **OneLiners** with four static functions described below. Try to implement them as 'one-liners', i.e. in such a way that each of them consists only of one **return** statement.

- `public static boolean hasTwoRoots(double a, double b, double c)` returns **true** if, and only if, the equation $ax^2 + bx + c$ has exactly two different solutions, and **false** otherwise;
- `public static boolean monot(double a, double b, double c)` returns **true** if, and only if, the numbers `a`, `b` and `c` constitute a strictly increasing sequence (each subsequent term is bigger than the previous) or a strictly decreasing one, and **false** otherwise;
- `public static double maxEl(double a, double b, double c))` returns the maximum value of the numbers `a`, `b` and `c`;
- `public static int numPos(double a, double b, double c)` returns the number of positive values among `a`, `b` and `c`.

For example, the following program,

download *OneLiners.java*

```
public class OneLiners {
    public static boolean hasTwoRoots(double a,
                                      double b, double c) {
        // ...
    }
    public static boolean monot(double a,
                                double b, double c) {
        // ...
    }
    public static double maxEl(double a,
                               double b, double c) {
        // ...
    }
    public static int numPos(double a,
                             double b, double c) {
        // ...
    }

    public static void main(String[] args) {
        double a = 2, b = 3, c = 1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): hasTwoRoots? " +
                hasTwoRoots(a,b,c));
```

```
        a = 0; b = 2; c = 1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): hasTwoRoots? " +
                hasTwoRoots(a,b,c));

        a = 2; b = 1; c = -1;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): monot? " +
                monot(a,b,c));
        a = 2; b = 1; c = 2;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): monot? " +
                monot(a,b,c));

        a = 2; b = 1; c = 2;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): maxEl = " +
                maxEl(a,b,c));
        a = 2; b = 2; c = 4;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): maxEl = " +
                maxEl(a,b,c));

        a = -2; b = 1; c = -3;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): numPos = " +
                numPos(a,b,c));
        a = -2; b = 1; c = 3;
        System.out.println("(a, b, c,)=(" + a + ", " +
                b + ", " + c + "): numPos = " +
                numPos(a,b,c));
    }
}
```

after implementing the functions, should print

```
(a, b, c,)=(2.0, 3.0, 1.0): hasTwoRoots? true
(a, b, c,)=(0.0, 2.0, 1.0): hasTwoRoots? false
(a, b, c,)=(2.0, 1.0, -1.0): monot? true
(a, b, c,)=(2.0, 1.0, 2.0): monot? false
(a, b, c,)=(2.0, 1.0, 2.0): maxEl = 2.0
(a, b, c,)=(2.0, 2.0, 4.0): maxEl = 4.0
(a, b, c,)=(-2.0, 1.0, -3.0): numPos = 1
(a, b, c,)=(-2.0, 1.0, 3.0): numPos = 2
```

Deadline: May 24 (inclusive)

Put yout Java file(s), and only Java files, in a directory the name of which is your surname (without Polish or any other non-ASCII characters). Names of Java files are arbitrary, although of course they should correspond to names of classes you created. Zip the whole directory ("from above" — not *just the files inside it). Then drop the zip file created in this way into folder "Tasks / Task_XX" of the GAKKO system (where 'XX' is the task number).*