

# Object Oriented Programming

## TDA547, DAT170

Krasimir Angelov, tel. 073-8228-179

2014-08-22

The total number of points is 40. 20 points certainly guarantee a pass. 27p correspond to grade 4 and 32p to grade 5.

No other help materials are allowed, except an English Dictionary and the distributed help sheet. Write clean and readable Java code. Trivial syntax errors will be tolerated without affecting the grades. You don't have to comment your code unless if you really want to.

1. Read the following program:

```
public class Question1 {
    public static void printMe(int[] a) {
        for (int i = 0; i < a.length; i++) {
            System.out.print(a[a.length-i-1]);
        }
        System.out.println();
    }
    public static void main(String[] args) {
        int[] a = {4, 1, 0, 2};
        printMe(a);
    }
}
```

What will the program print when it is executed? (4p)

2. In this task we do simple array processing:

- Implement the method:

```
public static int findMax(double[] a)
```

Which returns the position of the biggest number in `a`. If the array is empty the method should return -1. You can assume that all numbers in the array are non-negative. (4p)

Example: `findMax(new double[] {4, 1, 0, 5})` should return 3.

- Write a `main` method which can be used to test the method `findMax`. If both `findMax` and `main` were methods in class `Question2`, it should be possible to run it like this:

```
> java Question2 4 1 0 5
The biggest number 5 is found at position 3.
```

i.e. it takes the array of numbers from the command line arguments and prints the biggest number as well as its position in the array. If the array is empty, you should print "No numbers found" (4p)

*Hint: Don't forget that the command line arguments are strings that you need to convert to numbers with `Double.parseDouble(..)`.*

3. A polynomial is an arithmetic expression like  $a_0 + a_1x + a_2x^2 \dots + a_nx^n$ . Every polynomial can be represented with an array where for instance `a[0]` will store  $a_0$ , `a[1]` ==  $a_1$ , etc. The length of the array is the highest degree in the polynomial plus one. You need to implement a class called `Polynomial` which does basic arithmetics with polynomials. It should have the following methods:

- A constructor:

```
public Polynomial(double[] a)
```

which initializes the class with the coefficients in the argument `a`.

- A method:

```
public double eval(double x)
```

which evaluates the polynomial for a given value of  $x$ , for example `eval(1)` for the polynomial  $1 + 2x + 3x^2$  should return 6.

*Hint: You can compute  $x^n$  by multiplying  $x$   $n$ -times with itself.*

- A method:

```
public String toString()
```

which returns the textual representation of the polynomial. Since in plain text we cannot use superscripts to express exponents, a

polynomial like  $1+2x+3x^2$  should be shown as `1 + 2*x + 3*x^2`. Note that the first and the second coefficient are special since we don't write `*x^0` and we write `*x` instead of `*x^1`.

(10p)

4. Implement the method:

```
public static String replace(String s, String s1, String s2)
```

which replaces every occurrence of the string `s1` in `s` with the string `s2`. For example:

```
replace("a black window isn't a window", "nd", "d")
```

should return `"a black widow isn't a widow"`. In the implementation you are not allowed to use the predefined method `replace` in the class `String`. Instead use a loop iterating over `s` and string concatenation to compute the final string. (10p)

*Hint: to get the  $i$ -th character from a string `s`, you can use the method `s.charAt(i)`.*

5. Implement the method:

```
public static List merge(List l1, List l2)
```

which merges the two lists `l1` and `l2` into one, i.e. if `l1` has the elements `x1, x2, x3, x4` and `l2` is `y1, y2` then the result should be `x1, y1, x2, y2, x3, x4`. We always alternate one element from `l1` with one element from `l2`. If one of the lists is longer than the other, then the remaining elements in the longer list are just appended. In the example `x3` and `x4` are just inserted without intervening elements from `l2`. (8p)

*Hint: you can use `Math.min(x,y)` or `Math.max(x,y)` to find the smallest or the biggest of two numbers.*