

# Demo 11 / 24.11

## Tasks (an example exam)

During the exam you don't have to remember exactly which subprograms of the Java's API library are available. In case you need a ready class, subprogram, or method you only need to explain the functionality of it and with which parameters it works. Programming tasks must be commented in JavaDoc style. Collect your points in the exam so that your score is 24 at the maximum. It would be sensible to demand you to select one of the tasks 1-2 or make the task 5 compulsory.

The field "Mallitentin tehtävien tekemiseen kulunut aika" in Korppi is reserved for the time it took to carry out this example exam you are reading. Add the time for each task separately. The purpose is to help us in planning the final exam. In the final exam you will have 4 hours time for completing your answers and you have a permission to take one double-sided A4 filled with your own markings.

- 1-2. **Create a program:** The main program calls the subprogram (function) which finds out and returns the second largest element of an integer array. The subprogram IS NOT PERMITTED to sort the array. Think of noteworthy special cases and determine how to handle them.

```
For example an array: 3 1 -7 9 15 8
The subprogram returns the value 9 (doesn't not print it)
and subprogram prints:
The second largest element of the array 3 1 -7 9 15 8 is 9
```

The printing of the array is done with an own subprogram which has to be created as well. You are allowed to use only the local variables in the main program, the parameter variables of the subprogram and the local variables of subprograms. (12p)

3. Draw a picture of the objects `Line` and `RPoint` and their relations when 3 ascending stairs are drawn in the main program.

```
package demo9;
import fi.jyu.mit.graphics.Line;
import fi.jyu.mit.graphics.RPoint;
import fi.jyu.mit.graphics.Window;

public class Stairs {

    public static RPoint stair(Window window, RPoint start) {
        RPoint mid = new RPoint(start.getX(), start.getY()+1);
        RPoint end = new RPoint(start.getX()+1, start.getY()+1);
        window.add(new Line(start, mid));
        window.add(new Line(mid, end));
        return end;
    }

    public static void main(String[] args) {
        Window window = new Window();
        window.scale(0, -1, 10, 10);
    }
}
```

```

        RPoint end = new RPoint(0,0);
        end = stair(window,end);
        end = stair(window,end);
        end = stair(window,end);
        window.showWindow();
    }
}

```

The creation of the `Line` object stores the references of the `RPoint` objects passed as parameters. (6p)

4. In your own words and examples, explain what happens when a subroutine is called. Think of a function-like subroutine with at least a few parameters. When does each parameter get born and when do they disappear. What is the difference between basic variables (`boolean`, `char`, `int`, `double`) and object variables. (6p)
5. Write a subroutine `remove(line, letter)` for which the following ComTest tests work. The types have been left for you to figure out. Also how to implement `<-`. Write a small main program to test the subroutine.

```

TYPE line <- "cat sits in a tree"; // initialize line
remove(line, ' ') === 4; line.toString() === "catsitsinatree";
remove(line, 'a') === 2; line.toString() === "ctsitsintree";
remove(line, 't') === 3; line.toString() === "csisinree";
remove(line, 'e') === 2; line.toString() === "csisinr";
remove(line, 's') === 2; line.toString() === "ciinr";
remove(line, 'x') === 0; line.toString() === "ciinr";

```

6. Answer the questions with a few lines of explanation and/or examples (pick up to 6p worth of subtasks):

a) How does one compile and run a Java program using the command line? What if the program needs to implement a library? (2p)

b) Calculate (include calculations) the decimal values of the following binary numbers. What assumptions did you make? (2p):

```

0100 0101
0011 1111
1000 0001

```

c) Let's assume a two's complement notation for negative 8 bit integers. What would be the notation of minus five (-5) in such a system? (1p)

d) The binary decimal number `1001.1110` as a base ten number. (1p)

e) How is an `ArrayList` better than a regular table? How is it worse? (2p)

f) What is the difference between the classes `String` and `StringBuilder`? (1p)

g) How much is `34/7` in Java. And how much is `34%7` in Java. (1p)

