Demo 9 / 10.11

Tasks

1. **Ville:** Run Ville web application (look: https://trac.cc.jyu.fi/projects/ohj1/wiki/villeEn) and carry out task round: Recursion. Scale the amount of tasks done to the range [1,0] and round the point you get with precision of 0.2. (An example: if you did 120/195 tasks (=0.615) mark 0.6 as you final point. If 140/195 (=0.72) mark 0.8.)

2. **Objects**: Get the old subprogram `Stairs` and change it so that the following main program works:

```
public static void main(String[] args) {
    Window window = new Window();
    window.scale(0,-1,10,10);
    RPoint next = new RPoint(0,0);
    next = stair(window,next);
    next = stair(window,next);
    next = stair(window,next);
    next = stair(window,next);
    next = stair(window,next);
    next = new RPoint(next.getX()-1,next.getY());
    next = stairDown(window,next);
    next = stairDown(window,next);
    next = stairDown(window,next);
    next = stairDown(window,next);
    next = stairDown(window,next);
    window.showWindow();
}
```

   Hint: see the Graphics library documentation of the class `Rpoint`. So the idea is to give the stair a `point` object instead of two real numbers. When the stair draws itself it returns the point where it ended. This way it's easy to draw the next stair.

3. **Loops**: Using the previous write the necessary subroutines so the following main program works (if n=5 the picture would look the same as before):

```
public static void main(String[] args) {
    int n = 50;
    Window window = new Window();
    window.scale(0,-1,2*n,n+10);
    RPoint next = new RPoint(0,0);
    next = stairs(window,next,n);
    next = new RPoint(next.getX()-1,next.getY());
    stairsDown(window,next,n);
    window.showWindow();
}
```

4. **Lists:** Use the program CalculateALetters.java as an example and create a subprogram which finds the longest word of a list. The main program prints the longest word and removes ALL the occurences of it from the list.

5. **Lists:** Add a subprogram `copy(words, length)` to the answer of the previous task. This subprogram returns a new list in which is copied all the words from the original list that have the same length as the given `length` parameter.

6. **Files**: Using the subroutine from the previous task write a program that gets a file name from the main programs argument and then prints it so that only lines of the same length as the longest are printed. Ease-off: If you can't get the file open otherwise, you can give the whole file path as a parameter to the program.

B1. Familiarize yourself with the ComTest and learn how to use it. Write your own tests and run them with the subprograms in tasks 4 and 5.

GURU-tasks

G1-2. Write a program that reads all the files given in the command line parameters, sorts the result and removes all lines that are the same. Lines are counted as same if they have the same characters without regard to letter capitalization. Finally it should print the lines. This could be used for example if one had several posting lists and wanted to combine them into one without multiple postings of the same message. Comprise the program of small enough parts and test each subprogram with `ComTest`. Use ready-made lists and their methods as much as possible.