Demo 2 / 22.9

Tasks

1. **Information presentation on a computer:** The binary numeral system has two numbers: zero and one. This is called a bit (binary digit). The least meaningful or last bit signifies the amount one, the second to last signifies two, the third four and so forth, going along the exponents of two (8, 16, 32, 64, ...). The amount a binary number signifies can be discerned by adding together the decimal numbers signified by the bits. For example, the binary number 1000101 as a decimal number would be:

   1*64 + 0*32 + 0*16 + 0*8 + 1*4 + 0*2 + 1*1 == 69

   Translate the following binary numbers into the more familiar decimal(base-10) system:

   ```
   10010
   1001110001
   1000000000000000
   1010101010101011
   10000000000000001
   10000000000000010
   10000000000000011
   ```

   In modern computers it is typical that memory is divided into eight bit long sections or bytes. What is the largest amount you could save in a space of two bytes (16 bits) not minding negative numbers? What if you had to be able to represent positive and negative numbers: what would then be the largest and the smallest number possible to be represented in a two byte space?

2. **Algorithm:** Write an algorithm (list of directions) in English (not Java) describing how to find a certain name in a phone book. Also write an algorithm to find the owner of a certain number in a phone book. Which of the algorithms is faster and why? (Read: http://en.wikipedia.org/wiki/Algorithm or http://simple.wikipedia.org/wiki/Algorithm).

3. **Algorithm:** For this task, don't use Java or any other programming language. Describe in English or at most in ”pseudo-code” how you would divide the following tasks into smaller pieces. Then divide each piece into even smaller pieces. Use natural expressions for repeating and conditions ("if ... then ... otherwise ..."):

   ```
   a) Doing the dishes
   b) Making coffee
   c) Directions to return this demo task
   ```

4. **Subroutines:** Fill in the following program `Stairs.java` so that it works as the comments say it does:

   ```
   import fi.jyu.mit.graphics.EasyWindow;

   /**
    * The program draws five steps starting from the coordinate (0,0)
    * and ending in the coordinate (5,5)
    * @author // Fill in
   ```

```
 * @version // Fill in
 */
public class Stairs {

    /**
     * This subroutine draws one rising stair in the window
     * starting from (x,y) and ending in (x+1,y+1)
     * <pre>
     *    |-------(x+1,y+1)
     *    |
     *    |
     *    |
     *   (x,y)
     * </pre>
     * @param window
     * @param x starting x of the stair
     * @param y ending y of the stair
     */
    public static void stair(EasyWindow window,double x,double y) {
        // Fill in
    }

    /**
     * @param args Not in use
     */
    public static void main(String[] args) {
        EasyWindow window = new EasyWindow();
        window.scale(0,0,10,10);
        stair(window,0,0);
        stair(window,1,1);
        // Fill in
        window.showWindow();
    }

}
```

5. **Subroutines**: continue the program of the previous task by creating a subroutine called `stairDown` into it. Add proper callings for that subroutine too. If everything went fine you should be able to write in good conscience following comments into the beginning of the program:

```
* After that descending stairs are drawn starting from point (4,5)
* and ending in (9,0)

You will get the following picture after successful completion:
          _ (5,5)
        _| |_
      _|    |_
    _|       |_
  _|           |_
 |               |
(0,0)           (9,0)
```

6. **Subroutines:** create a program which draws in a window four isosceles triangles with a height of one unit and width of two units. A hint: make a subroutine called `triangle` to which you pass needed information as parameter.

B1. Begin this task by opening the Alice's ice skating example. Begin the tutorial and exit it right after opening it. Look at the trees in the background. Put two of them to arrive to the skater so that the first comes in 2 seconds and falls down 45 degrees to the left right after the second tree has arrived. The second tree will fall down as well, but 45 degrees to the right. So, in this task you have to make things collide.

As an answer you have to return a text file with the "program lines" you made and the name of the file is in your decision. Example of one such line in Alice:

```
IceSkater.do simple spin
```

You will find information on Alice here: http://trac.cc.jyu.fi/projects/ohj1/wiki/AliceEn/

GURU-task

G1-2 First, familiarize yourself with the example program `CarSample.java` from the library: http://users.jyu.fi/~vesal/kurssit/ohj1/graphics/ . Redo tasks 4,5 and 6 so that you create classes `Stair`, `StairDown` and `Triangle`. With these classes you draw a staircase (naturally with a for loop now) and 4 triangles in a window. One of the triangles has to be rotatable.