Java lab 1 - 06 october 2009

The answers to the exercices below are due before *09 October 2009*. The procedure to send the exercices is:

1. Create a directory named `yourname-lab1`.
2. Copy all the `.java` source files to this directory.
3. Issue following command in a shell : `tar czvf yourname-lab1.tar.gz yourname-lab1/` to create a tarball.
4. Send this tarball to `pablo@sifflez.org` with subject `yourlastname-lab1`.

I) Java toolchain

You will put each java class you write in its own file.

For a class named 'Example' you will use file `Example.java`. To compile this class to bytecode you will execute:

```
$ javac Example.java
```

If no errors are found this will produce a bytecode file named `Example.class`.

To run a class that contains a `main` method you will issue below command in the directory containing `Example.class`:

```
$ java Example
```

II) Prime Numbers

Exercice 1 [**Numbers**]
Write a class `Numbers` with a main method that prints the numbers from 1 to 100.

Exercice 2 [**Primes - Naïve**]

1. A naïve method for determining if a number n is prime is checking that all the numbers smaller than $\sqrt{n}$ are not divisors of n.

   Add a static method "`boolean isPrime(long n)`" that implements this algorithm in the class `Numbers`. You might need to use the method `double Math.sqrt(double n)`.

2. Modify `Numbers` so that it prints all prime numbers smaller than 100.

III) Turtle graphics

In 1967 W. Feurzeig and S. Papert created a programming language called *Logo* to teach programming. This language allowed to move a turtle on a graphical screen to produce all kind of pictures.

You are going to use a java package called `TurtleGraphics` to make some pictures using java. The package `TurtleGraphics` contains two classes:

- `class Sheet`
  The class Sheet implements a graphical window that we can draw onto. It provides the following methods:
  - `Sheet(int width, int height)`, the constructor, you must provide the dimensions of the graphical window.
- `class Turtle`
  - `public Turtle(Sheet sheet)`, the constructor, you must provide a sheet for the turtle to draw on.
  - `void turn(double degrees)`, makes the turtle turn.
  - `void advance(double steps)`, makes the turtle advance.
  - `void penDown()` and `void penUp()`, change the pen state, when you move the turtle and the pen is down it will leave a trail on the sheet.
  - `void setPenColor(Color color)`, changes the pen color.

For example the following program draws a Triangle:

```
import java.awt.Color;
import TurtleGraphics.Sheet;
import TurtleGraphics.Turtle;

public class AdvancedTurtle extends Turtle {
  public AdvancedTurtle(Sheet s) {
    super(s);
  }

  public void triangle(double side) {
    advance(side);
    turn(120);
    advance(side);
    turn(120);
    advance(side);
  }

  static public void main(String args[]) {
    Sheet sheet = new Sheet(300,300);
    AdvancedTurtle t = new AdvancedTurtle(sheet);
```

```
    t.penDown();
    t.triangle(50);
  }
}
```

Exercice 3 Add a method `square` to AdvancedTurtle that draws a square of given side.

Exercice 4 Add a method `circle` that draws a circle of given radius.

Exercice 5 [Koch's Snow Flake]
On 1904 Helge Von Koch described the fractal curve obtained by the following algorithm:

Koch's algorithm:

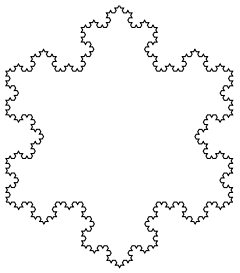
1. take a segment.

2. divide it in 4 new segments of *equal length* according to the following picture:

3. iterate this procedure for each new segment:

iteration 2 iteration 3

1. Add a method `koch(double length)` that draws a Koch's curve of given length.
2. Add a method `snowFlake(double length)` that draws a Koch's snowflake, obtained by applying the Koch's algorithm to the three sides of an equilateral triangle:



IV) Back to prime numbers

Exercice 6 [**Eratosthenes sieve**]
Eratosthenes was a greek mathematician, geographer and astronomer which calculated using

the elevation of the sun what we believe to be the first measure of the earth circumference. He also produced an efficient method to compute prime numbers smaller than N, which we explain below.

Eratosthenes algorithm:

1. Make a list of numbers from 2 to N, call it numbers.
2. Take an empty list and call it primes.
3. Take the smaller number k in list numbers.
4. Append k to list primes.
5. If k^2 is bigger than N, stop and append all the remaining values in numbers to primes.
6. Remove all the multiples of k from numbers.
7. Start again at 3.

Implement this algorithm in java using arrays for the lists. Here are some guidelines to help you, they are not mandatory:

- Create two arrays:
  - `numbers` of size `N-1` and with `boolean` type
  - `primes` of size $N - 1$ and with `long` type.
- `numbers` is a boolean array, it allows us to handle the list of numbers:
  - if cell `k` is `true` that means that `k` is still in list numbers
  - if cell `k` is `false` that means that `k` has been removed from list numbers
- `primes` is used to store the primes when we find them.