Exercise 4. *Deadline: 9 November 2009.*

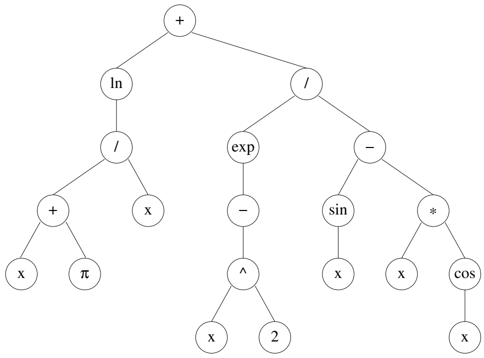
THE JAVA PROGRAMMING LANGUAGE

EXPRESSION TREES

University of Wrocław
Institute of Computer Science *Paweł Rzechonek*

Exercise

An expression tree is a rooted tree used to represent a mathematical expression. Each internal node in a tree is an arithmetic operation or mathematical function, and all leaves are operands (constants or variables). For example, the following tree corresponds to the expression $\ln\left(\frac{x+\pi}{x}\right) + \frac{e^{-x^2}}{\sin x - x\cos x}$.



Define an abstract base class `Expression` for representing arithmetic expressions. The class should contain an abstract method `calc()` that calculates the value of the expression.

```
public abstract class Expression implements Comparable<Expression>
{
    public abstract double calc ();
};
```

Next, define derived classes `Real` (for a constant real number), `Var` (for a variable), and classes for arithmetic operations (negation, addition, subtraction, multiplication, division, etc.), comparisons (less than, equals to, greater than, etc.) and different mathematical functions (sin, cos, log, power, ln, exp, abs, min, max, etc.). Design a class hierarchy for those operations. Place a set of variables of type `HashMap<String,Double>` into the class `Var` as a static data member.

Finally write a short program, which will test your implementation of mathematical expressions. Create and calculate the following expressions (first set the variable x to 1):

```
3+5
7*11+2.72/3.14
abs(x-4)
(3*11-1)/(7+5)
((x+1)*x)/2!=0
sin(pi/8)
exp(-x)
```

Implement the method `toString` in each class.