Exercise 4 Deadline: 28 May, 2020.

C++ PROGRAMMING LANGUAGE

Expression trees

University of Wrocław
Institute of Computer Science

Paweł Rzechonek

Prologue.

An *expression tree* is a representation of algebraic expressions arranged in a tree–like data structure. In other words, it is a tree with leaves as operands of the expression and nodes contain the operators. Expression trees are mainly used for analyzing, evaluating and modifying expressions.

Task (similar to the task on Java course).

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

Next, define the classes that inherit from the `expr` class, which will represent the operands and operators. The operands are: numbers (floating–point value of type `double`), variables (the variable has a name of type `string` and reference to variable of type `double`; variables are stored in a set of variables) and constants (constant has a name of type `string`, which is associated with a fixed value). The operators represent arithmetic operations (addition, subtraction, multiplication, division, exponentiation, change of sign to the opposite, etc.) and selected mathematical functions (sine, cosine, logarithm, etc.). In derived classes, override the method `calc()` and define stream operator for writing `<<`.

Finally write a short program, which will test your implementation of algebraic expressions.

Hint.

Remember variables in the association set of type `vector<pair<string,double>>`. Declare this set as a static field in the `variable` class.

Notice.

Divide the program into header and source files. Place the declarations all classes inthe `calculation` namespace. Put the `main()` function in a separate source file.