Wrocław, 16.11.2015

Assignment #5
Calculator

1. Analyze carefully the module EvalExpression:

```
#ifndef EVALEXPRESSION
#define EVALEXPRESSION
int isValidExpression(const char * str);
int getOperator();
float getFstOperand();
float getSecOperand();
float getExprValue();
#endif // EVALEXPRESSION
```

```
#include <stdio.h>
#include "EvalExpression.h"

static float f1, f2;
static char op;
int isValidExpression(const char * str)
{
    int  res;
    char ops[10];
    res=sscanf(str, "%f %s %f", &f1, ops, &f2);
    op=ops[0];
    return(res==3);
}
int getOperator()
{
    return(op);
}
float getFstOperand()
{
  return(f1);
}
float getSecOperand()
{
  return(f2);
}
```

2. **Extend the module EvalExpression so that it:**
   - Accepts only proper operands that is: + * - / and ^ (power)
   - Implements the function getExprValue.

3. **Implement the filter CalcFilter that:**
   - Expects that each line contains an expression accepted by the function isValidExpression.
   - The filter accepts one switch in the form of /E or /V
   - Ignores all not valid lines and print the value their values (switch /V)
   - Prints all not valid lines and ignore the rest (switch /E)

4. **Implement the program InterCalc that:**
   - Reads lines from standard input and evaluates them.
   - In case of an error a message: “expression error” is printed.
   - And of file or line with ‘exit” ends the work of the program.

Andrzej Siemiński