

Expression Calculator

Project 4

201220130
Chang Shuhao

C A T E G O R Y

01

需求分析

02

文件结构

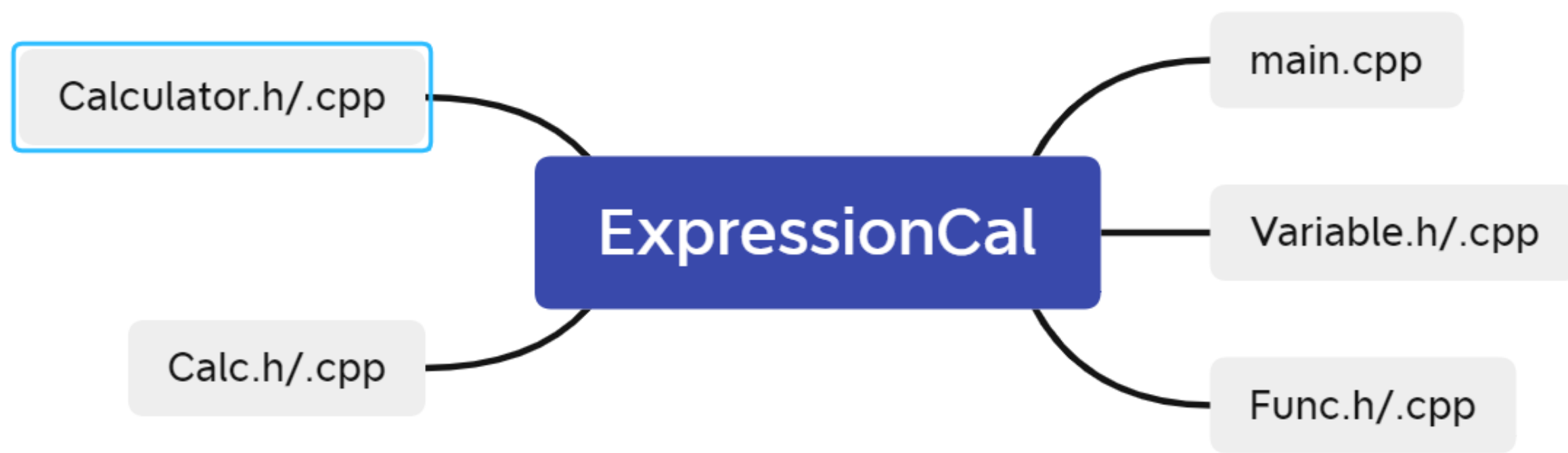
03

数据结构

04

拓展功能

- 实验环境：Linux
- 实验内容：实现一个表达式计算器
 - 判断输入是否合法
 - 合法，输出计算结果；不合法，输出表达式存在错误
- 表达式中的符号
 - 变量符号：由[a-zA-Z]+表示
 - 常量符号：常量 \in 有理数
 - 二元运算符：+、-、*、/
 - 函数：形如f(·, ·, ...)
 - 括号：（、）
- 输入：
 - 定义变量：Var [变量名] = [表达式]
 - 定义函数：Func [函数名](变量名) = [表达式]
 - 变量赋值：[变量名] = [表达式]



Variable.h/.cpp

表示变量和常量的Class，用isConst来判断是否是常量，成员变量val表示值，常量是一个只包含数字的字符串，变量是一个表达式（不进行计算）
包含成员函数eval

```
//  
// Created by csh on 2021/6/4.  
//  
  
#ifndef EXPCALC_VARIABLE_H  
#define EXPCALC_VARIABLE_H  
#include <string>  
  
using namespace std;  
class Variable {  
public:  
    string eval();  
    bool isConst;  
    Variable(string name, string val);  
  
    const string &getVarName() const;  
  
    void setVarName(const string &varName);  
  
    const string &getVal() const;  
  
    void setVal(const string &val);  
private:  
    string varName;  
    string val;  
};  
  
#endif //EXPCALC_VARIABLE_H
```


Function.h/.cpp

表示函数的Class，内有map<int, string>arguments表示<index, 表达式>，map<string, int>trans表示形参名对应的index，body表示函数体，name表示函数名包含成员函数eval

```
#ifndef EXPCALC_FUNC_H
#define EXPCALC_FUNC_H
#include <iostream>
#include <vector>
#include <sstream>
#include <regex>
#include <map>
#include <queue>
using namespace std;
class Func {
public:
    string eval(string args);
    map<string, int> trans;
    map<int, string> arguments;
    string body;
    string name;
    int arguments_size;
};
```

Calculator.h/.cpp

含有static变量funcs, vars, 同时有相应函数来负责读入, 创建函数、变量, 以及计算值, 含有成员函数doCal

```
#ifndef EXPCALC_CALCULATOR_H
#define EXPCALC_CALCULATOR_H

#include "Func.h"
#include "Variable.h"
#include "Calc.h"
#include <iostream>
#include <string>
#include <cstring>
#include <vector>
#include <algorithm>

using namespace std;

enum ACTION {
    CREATE_VARIABLE, SHOW_VALUE, ASSIGN_VALUE, CREATE_FUNCTION
};

class Calculator {
public:
    // Calculator();
    static vector<Func*> funcs;
    static vector<Variable*> vars;
    // static map<string, int> operatorRank;
    void doSth();
    int analyseLine(string& line);
    void create_variable(string& line);
    void show_value(string &line);
    void assign_value(string &line);
    void create_function(string &line);
    bool check(string line);
};
```

Calc.h/.cpp

一个表达式计算器，能够计算出仅含数字的表达式的值，但是只支持+、-、*、/

```
class Calc{
private:
    char Exp[MAXN];          //表达式
public:
    const char *getExp() const;

    double getAns() const;

    void setAns(double ans);

    void setExp(char* exp);

private:
    int NextB[MAXN];
    double Ans;
    void DelandLower(char *str);          //删除空字符 转化为小写
    bool Check(char *str,int & len);
    bool CheckCh(const char *str,int pos); //检查字符
    bool Is_Num(char c);                  //是否为数字
    bool Operat(char c);                  //是否为运算符
    bool CheckError(const char *str,int len);
    bool CrectB(const char *str);          //检查括号匹配
    bool Equal(double a,double b);         //判断浮点数相等
    int Prio(double x);                    //符号优先级判断
    double hash(char c);                   //符号到浮点型映射
    double GetV(const char *str,int st,int ed); //区间求值
public:
    void Input(){cin>>Exp;}
    void Output(){printf("%.2f\n",Ans);}
    bool Cac();
};
```


main.cpp

创建计算器Calculator c，while(1)循环，循环中每次调用c.doSth()

```
#include <iostream>
#include "Calculator.h"

using namespace std;

int main() {
    Calculator calculator;
    while(1) {
        calculator.doSth();
    }
    return 0;
}
```

更加好看的报错

```
cs@LAPTOP-KQ946URK:~/projects/c++/Expression-Evaluator|main ⚡ ⇒ ./main
ExpCal > Var x = 1
ExpCal > y
ERROR: VARIABLE y DOESN'T EXIST
ExpCal > x = f(x,y)
ExpCal > x
ERROR: FUNCTION f DOESN'T EXIST
ExpCal > x = 1 / 0
ExpCal > x
ERROR: DEVIDE ZERO
ExpCal > x = 1..3
ExpCal > x
ERROR: VARIABLE 1..3 DOESN'T EXIST
ExpCal > █
```

```
ExpCal > x = (1 + 2))
ERROR: PARENTHESIS DON'T MATCH
ExpCal > █
```

谢谢观看

— THANKS —

Project 4