



北京大学

大作业：函数式计算器

姓名：董欣然

学号：1900013018

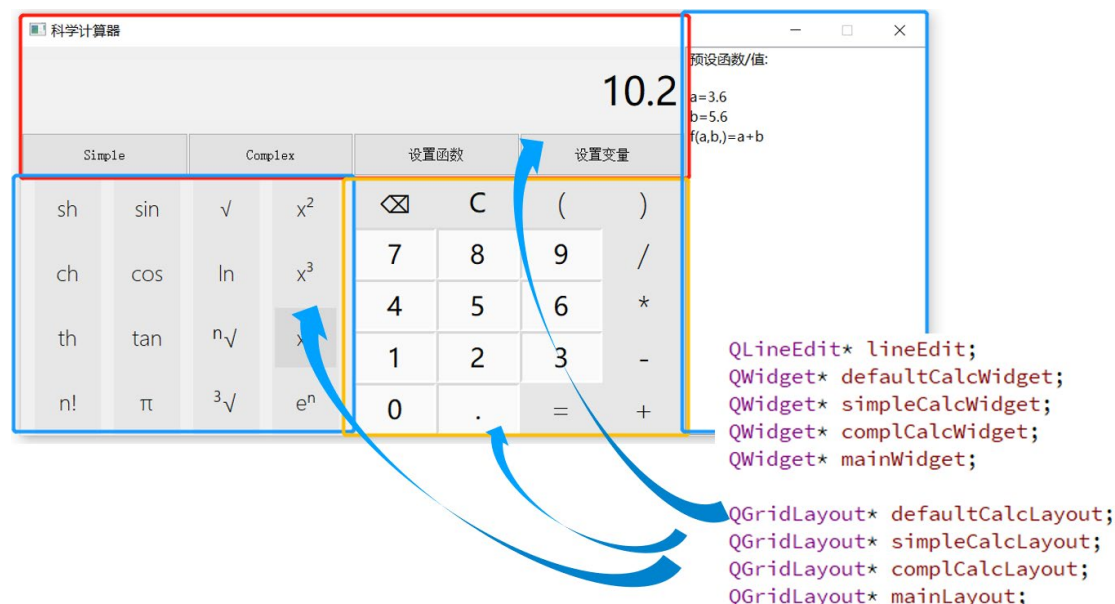
课程：C++语言程序设计

任课教师：邓习峰

概览

作者的大作业项目是函数计算器。函数计算器实现了四则运算，预定义函数，预定义变量，科学模式运算，程序员模式运算，错误操作下安全运行处理。

计算器分为四个基本组件，分别是计算器输入框和模式切换组件 (defaultCalcLayout)，标准计算器键盘组件 (simpleCalcLayout)，科学计算器键盘组件 (complCalcLayout) 和自定义值/函数显示栏 (historyWidget)。效果如下

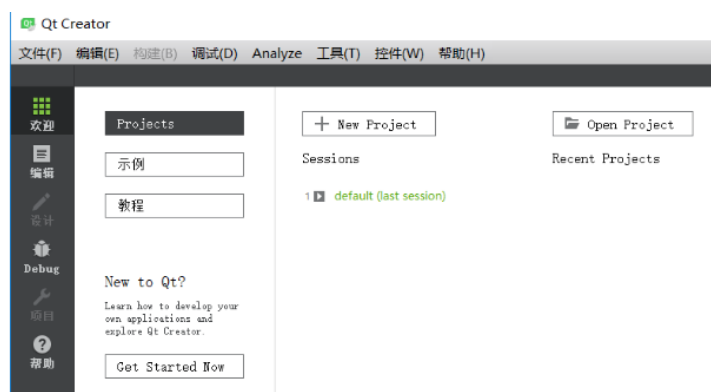


程序运行方式说明

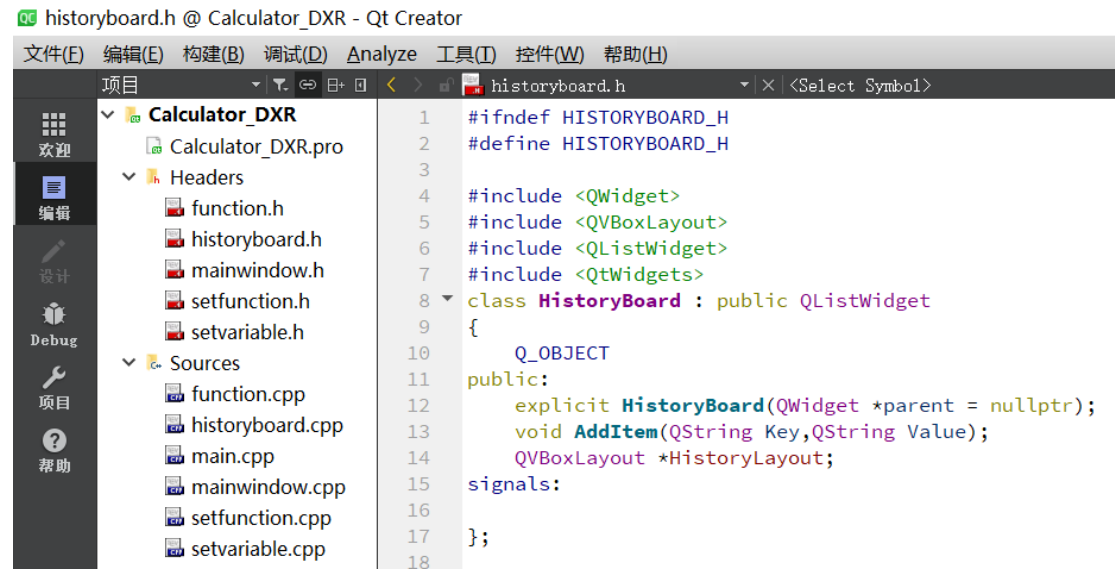
作者使用操作系统 windows10，电脑型号是联想小新 air15。使用 C++ 完成，使用 QT 实现可视化界面。

操作步骤说明

1. 将作者压缩包中的 Calculator_DXR 解压至桌面，Calculator_DXR 文件夹中包含所有源代码和项目文件 Calculator_DXR.pro
2. 下载 QT 软件，作者使用的是 [qt-opensource-windows-x86-5.14.0.exe](https://blog.csdn.net/zhizhengguan/article/details/107567449) 软件下载教程可参考 <https://blog.csdn.net/zhizhengguan/article/details/107567449> 下载时勾选“Qt Creator”组件。作者使用的是 Qt Creator4.11.1 (community)
3. 下载完成后，打开 Qt Creator，效果如下



4. 点击左上角“文件”->点击“打开文件或项目”->点击 Calculator_DXR 文件夹中的 Calculator_DXR.pro ->点击“打开”，此时 Qt Creator 将自动构建项目，效果如下



5. 右击 Qt Creator 中的 Calculator_DXR 文件夹，点击“运行”，此时便可呈现计算器。效果如下



类文档说明

作者一共编写了五个源文件（头文件）来实现计算器的各个功能。分别为

1. mainwindow.h
2. function.h
3. SetFunction.h
4. SetVariable.h
5. HistoryBoard.h

以下将对这些文件逐个展开介绍

1. 主界面构造 mainwindow.h/mainwindow.cpp

该程序声明了类 MainWindow，该类的作用是联合调度所有部件，生成一个程序员计算器。类 MainWindow 有以下成员函数：

`void CreateDefaultCalcWidget();` 构造计算器输入框和模式切换组件
`void CreateSimpleCalcWidget();` 构造标准计算器键盘组件
`void CreateComplCalcWidget();` 构造科学计算器键盘组件
`void SpawnSimpleMode();` 切换为标准计算器模式
`void SpawnComplMode();` 切换为科学计算器模式

以下为接收信号的槽函数

public slots:

`void SwitchComplMode();` 捕获“切换为科学计算器模式”信号
`void SwitchSimpleMode();` 捕获“切换为标准计算器模式”信号
`void ModifyFunction();` 设置函数
`void ModifyVariable();` 设置变量
`void NumberClicked();` 接收数字按钮的信号
`void UnFnClicked();` 接收一元运算按钮的信号
`void BinFnClicked();` 接收二元运算符按钮的信号
`void DotClicked();` 接收小数点按钮的信号
`void ClearInput();` 清空输入框
`void BackSpace();` 撤回输入
`bool Equals();` 表达式求值

2. 计算表达式 function.h function.cpp

该程序声明了类 Function，该类会对给定的表达式进行求值，表达式中可包含自定义变量、自定义函数、三角函数、双曲函数、位运算、四则运算等。

类 Function 包含以下成员变量：

`vector<Variable> variableList;` 自定义变量列表
`vector<Function> functionList;` 自定义函数列表
`stack<string> opStack;` 操作符栈
`stack<double> numStack;` 数字栈
`stack<double> temporaryStack;` 函数参数栈

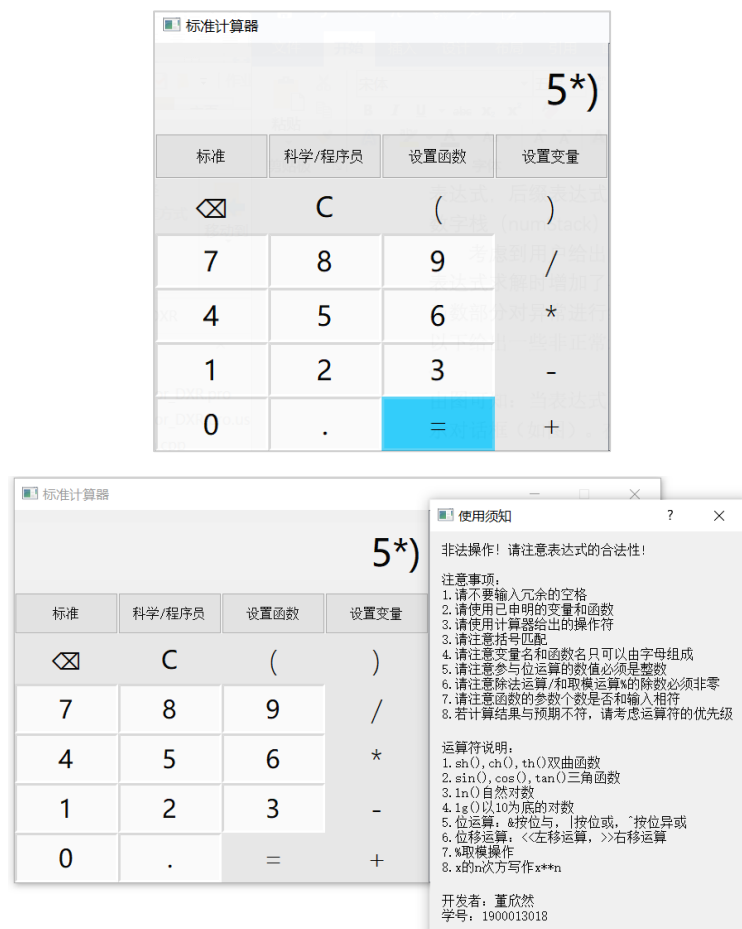
类 Function 包含以下成员函数：

```
double readDigit();    读入数字
string readLetter();   读入字符串
bool isDigit();        判断是否是数字
bool isLetter();       判断是否是字母
bool isFunc(string ss); 判断是否是函数
bool isDualOp(string s); 判断是否是二元运算符
bool isBinaryOp(string s); 判断是否位运算
bool isMathOp(string s); 判断是否是科学计算器运算
void calcDualOp(string c); 进行二元运算
double calcMathOp(string, double number); 进行科学计算器函数计算
double newExpression(); 新建表达式
int getLevel(string s); 运算符优先级
double Expression2Value(); 表达式求值
```

function 类是实现表达式求值的核心函数，我设计的算法是：将中缀表达式转化为后缀表达式，后缀表达式的计算简化了函数和括号嵌套问题，可以使用运算符栈（opStack）和数字栈（numStack）来维护出后缀表达式的求解过程。

考虑到用户给出的表达式存在非法情况，为了避免非法运算导致程序崩溃的情况，我在表达式求解时增加了异常处理，针对各种会导致程序非正常运转的情况及时抛出异常，在主函数部分对异常进行统一处理，增强代码的鲁棒性。

以下给出一些非正常运转情况示例：



由图可知：当表达式不合法时，在鼠标点击或键盘按下“=”键、“回车”键或“设置变量”键时会弹出错误提示对话框（如图）。在获知可能出错的情况下，用户可以对表达式做出更改，计算器仍然可以正常运行。

3.“设置函数”组件 SetFunction.h/SetFunction.cpp

设置函数功能允许用户设置含有 0 至 3 个参数的表达式，表达式不限于四则运算，也可以包含科学计算器中的其他运算。

该程序中定义了 SetFunction 类，包含以下成员变量

```
QLineEdit *FunctionNameEdit; 函数名编辑框  
QLineEdit *ParameterNameEdit1; 参数名 1 编辑框  
QLineEdit *ParameterNameEdit2; 参数名 2 编辑框  
QLineEdit *ParameterNameEdit3; 参数名 3 编辑框  
QLineEdit *ExpressionNameEdit; 函数表达式编辑框
```

包含以下检测函数，当检测到用户输入不合法时，程序会抛出异常。

```
bool isLetter(char c);          判断 c 是否是字母  
bool checkString(string s);    判断定义的名称合法性  
bool check();                  判断定义的函数合法性
```

4. “设置变量”组件 SetVariable.h/SetVariable.cpp

该程序定义了类 SetVariable，其中实现了设置变量功能。其函数定义与 SetFunction 类似。不再赘述。

5.“历史显示栏”组件 HistoryBoard.h/HistoryBoard.h

该程序定义了 HistoryBoard 类，该类有成员函数 AddItem(QString Key,QString Value)用于显示新增的条目。当 SetVariable 和 SetFunction 抛出异常时，“历史显示栏”会提示此次设置不合法，并且忽略该设置行为。