

实验：简单计算器

2020 级图灵实验班 石依凡 2020202264

一、需求分析

用户输入一个命令语句或表达式语句，计算机给出相对应的结果。

a. 输入形式（详细说明文档见 README 文件）：

- 命令语句输入：*cmdExpr cmdArgs*
- 表达式语句输入：正常书写习惯

b. 输出形式：

- 正常输出：*Output [Index]*
- 错误输出：*ERROR_TYPE ERROR!*

c. 功能列举：

1. 具有交互界面或图形界面。
2. 用顺序表来完成任意同维度向量的计算，包括加法、减法、夹角余弦值等。
3. 使用顺序表来完成任意一元多项式的计算，包括加法、减法、乘法、导数(包括任意阶)等。
4. 使用链表来完成任意一元多项式的计算，包括加法、减法、乘法、导数(包括任意阶)等。
5. 四则运算表达式求值。操作符包括加('+')、减('-')、乘('*')、除('/')、幂('^')、左括号('(')、右括号(')'),而操作数则包括整数、浮点数等不同类型的数值。
6. 含单变量的表达式求值。变量可以是 C/C++ 的标识符。
7. 定义并运行简单函数。
8. 保留函数定义历史，并可以运行历史函数。
9. 函数的调用、复合。内置所有基本初等函数和常用函数。
10. 操作数的各种字面常量的处理（科学计数法、负数）。
11. 支持矩阵的运算：矩阵的加、减、乘、行列式的值等。
12. 其它未列出的功能。

二、概要设计

本项目使用面向对象编程（OOP）的基本方法，包含以下的类、模版类：

- **Stack <typename T>**: 基本数据结构（栈）模板类
- **ChainNode <typename T>**: 栈节点模版类
- **Function**: 自定义函数类
- **Expression**: 自定义表达式类
- **Operator**: 自定义运算符类
- **Matrix**: 自定义矩阵类
- **Determinant: public Matrix**: 自定义行列式类（继承 Matrix 类）
- **Vector**: 自定义 n 维向量（ n 次多项式）类

包含以下的结构体和枚举：

- **Node**: 向量节点结构体
- **Value**: 数值结构
- **MODE**: 程序模式枚举
- **TYPE**: 储存模式枚举
- **CalcStatus**: 计算状态枚举

包含两个非类定义代码文件：

- **main.cpp**: 主函数、主消息循环、主命令分析处理
- **test.cpp**: 内置测试点及自动评测系统

包含三个样例文本文件：

- **EXAMPLES**: 测试样例
- **EXAMPLES – Arithmetics and Special**: 测试样例补充（四则运算部分）
- **Function Examples**: 测试样例补充（函数部分）

包含一个 README 说明文件。

本项目使用 CMake 进行跨平台开发，使用 C++23¹作为 C++版本，使用 STL（非课程要求部分）优化。开发、测试内容基于 macOS Monterey, x86_64。文件编码为 UTF-8。

¹ 读者需要将 C++标准设置为 C++20 或更新版本，因为代码中使用了 `std::map` 的 `contains()` 方法，这一特性于 C++20 加入。

三、详细设计

主函数接受用户输入，将其解析为一行命令，之后发送给 `process()` 函数。
`process` 分析命令内容，判断命令种类，选择合适的分支语句，实现相对应的功能。

- 本项目中所有的字符串处理，均使用 `string` 类和 `stringstream` 类。
- 本项目对基础功能进行面向对象封装，通过向主函数提供接口，从而使逻辑框架清晰，维护方便。

关于每个类、模版类的设计，请读者自行阅读代码文件。

四、调试分析

将于上机课进行集中展示。

五、用户使用说明

见 README 和 3 个说明文本文件。

六、附录

代码请见附件。