

中山大学本科生课程实验报告

一个交互式多项式图形计算器的实现

15323032 李新锐

lixr26@mail2.sysu.edu.cn

2017 年 10 月 12 日

项目原创性声明

本人郑重声明：本人所呈交的软件代码和实验报告，是本人独立进行学习研究所取得的成果。除已经注明引用的内容外，不含任何其他个人或集体已经发表或撰写过的作品或成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的一切结果由本人承担。

作者签名: _____

日 期: _____

摘要

本实验目的在于实现一个多项式计算器程序。我成功完成了“将系统进一步升级，并用简单的命令语言进行多项式计算”这一要求。主要用到了基于栈的中缀表达式运算算法，线性表式多项式表示法、多项式加减算法和基于快速傅里叶变换（FFT）的多项式乘法算法。除此之外，还实现了多项式的长除法算法和图形界面，绘制函数图像，文件写入读取等额外功能。

目录

1 问题说明	4
2 设计和实现	5
2.1 基本架构	5
2.2 关键模块说明	5
2.2.1 主界面	5
2.2.2 作图界面	7
2.2.3 calc_interface 类	7
2.2.4 Polynomial 类	10
2.2.5 快速傅里叶变换多项式乘法	11
3 程序使用和测试说明	14
3.1 程序使用	14
3.1.1 安装说明	14
3.1.2 使用说明	14
3.2 四则运算测试	16
3.3 求值测试	17
3.4 求导测试	18
3.5 打印变量测试	19
3.6 作图测试	20
3.7 错误提示	22
4 总结讨论	24
5 致谢和参考文献	25

1 问题说明

本实验要解决的问题是设计一个多项式计算器，具体而言，可将问题细分为：

1. 设计多项式在计算机中储存的数据结构。
2. 设计多项式进行加减乘除、求导、判断相等操作的算法。
3. 设计与用户交互的输入输出形式。
4. 设计将用户的输入正确地转换为计算机内部的多项式运算的方法。

解决以上问题，可以实现出一个用户友好，具有实用价值的计算器程序。对于以上问题，解决方法分别是：

1. 将多项式的逻辑结构抽象做系数、指数二元组的线性表，用对应的物理结构进行储存。
2. 加法减法可以模仿手算多项式的方法，将相同指数的项相加减，指数不同的项与 0 相加减，再加入结果多项式中。乘法可以使用快速傅里叶变换算法在 $O(n \log n)$ 的复杂度内完成。除法可以模仿手算长除法的方法进行。但由于本计算器中支持的运算对象是多项式，故除法得到的分母是多项式的余数将被舍弃。
3. 使用图形界面，在一个窗口中可以有多个子窗口，分别用于输入语句， x 的值，输出，求值结果以及变量列表，能够方便用户使用，提高程序的实用性。
4. 首先判断用户输入的类型，对于简单的测试相等、给变量赋值的语句可以直接使用 C++ 的字符串处理功能进行处理。对于赋值的变量间运算的语句，要使用基于栈的中缀表达式运算算法。对于用户输入，判断输入的合法性贯穿整个处理过程之中，方法主要是基于 Qt 的正则表达式功能进行词法分析，并根据一些正确的输入要有的特征（例如测试相等语句中操作数个数应为 2 个）进行的简单的语法错误检查。

2 设计和实现

2.1 基本架构

如下图所示，本项目的基本架构分为以下三层，



图形用户界面层 本项目中，我通过 QtCreator 设计了 MainWindow、draw_figure_window、about 三个 ui 文件，Qt 从而生成了三个对应的窗口类，实现了程序的图形界面。主要用到了 Qt 中 QMainWindow、QDialog、QMessageBox 等窗口控件，QLineEdit、QTextBrowser、QWidgetView、QPushButton 等输入输出控件，以及这里用于绘制函数图像的的 QChartView 控件。

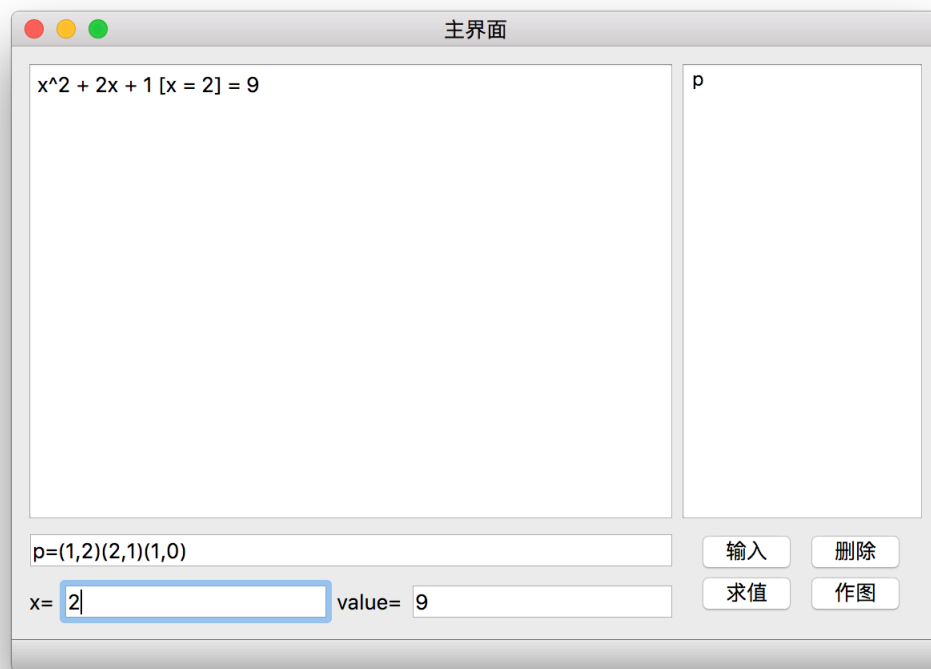
用户输入解析层 本项目中，用户输入解析层主要在 calc_interface 类中实现。这个类中主要是静态函数和静态数据成员。既提供了图形界面可以调用来处理用户输入的接口，也储存了反馈给图形界面的处理输入的结果有关的信息。

多项式代数库 本项目中，多项式在 Polynomial 类中实现，多项式储存在该类的 polynomial 成员中。该类重载了加减乘除、相等、输入输出等多种运算符，实现了求值求导等运算。提供了从数字/二元组（单个项）、字符串新建多项式的构造函数，插入新项、判断多项式是否为 0（即 polynomial 为空）、获取项数（polynomial 元素个数）等操作接口。

2.2 关键模块说明

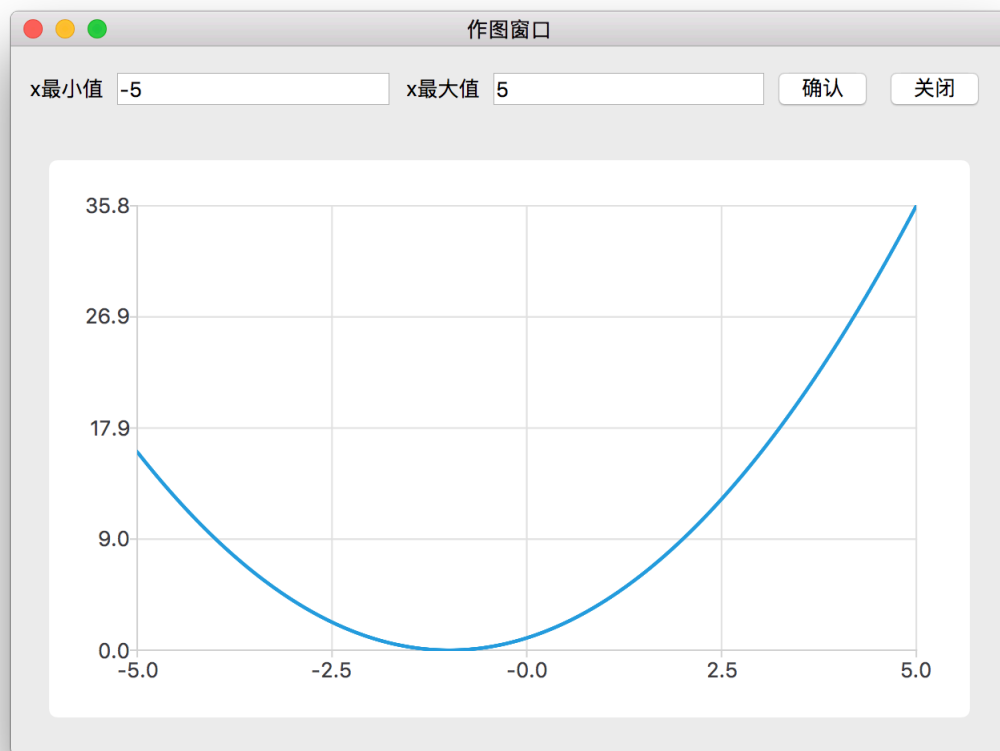
2.2.1 主界面

本节主要目的是通过一些示例说明图形界面层的工作方式，更详细的使用说明请见第四节。



如图所示：正输入赋值语句 $p = (1, 2)(2, 1)(1, 0)$ 的 QLineEdit 控件（输入按钮左侧）是主输入条，按下回车或输入后，将在主输入条上方的主输出窗口中输出运算结果。接受的语句形式符合实验说明文件中关于“简单的命令语言”的说明。右侧是变量窗，显示现存的变量。若未将当前结果赋给任何变量，则将保存在内置变量 `ANS` 中。选择变量后双击可加载之，点删除可删除该变量。下方有 x 的值输入条，输入后按回车或求值按钮，将在主输出窗和 `value` 输出条中显示求值结果。

2.2.2 作图界面



在主输入条中输入多项式函数解析式后，点击作图按钮，打开作图窗口，输入 x 的范围，点击作图，程序会通过计算一系列点的值，添加到 `QChat` 类中，即可在下方的 `QChartView` 控件中做出指定范围内的函数图像。菜单栏有文件，帮助两个选项，可将当前运算结果和变量值保存为数据文件，或打开之前储存的数据文件。点击帮助可查看使用说明。

2.2.2 主要类的具体实现说明

2.2.3 `calc_interface` 类

`process_input_main` 函数 它接收主输入条和 x 值输入条中用字符串形式输入的语句和 x 的值，调用 `doParse` 函数进行处理，根据下面所述的三个错误判断有关结构体的内容判断处理过程中是否发生错误，将处理结果存放在类型为 `process_result_t` 的结构体内，该结构体

储存处理结果，包括对主输入框内容的影响 (main_result)，对变量框的影响 (variable_result)，对求值框的影响 (eval_result)，返回给图形界面显示出来。

doParse 函数 该函数是实质对用户输入字符串进行处理的重要函数。首先重置有关处理结果和错误的静态成员信息。该函数的主要工作流程如下：

```

1  重置有关处理结果和错误信息的静态变量
2  将输入去空格化
3  若输入包含 "print"
4      判断输入类型为打印语句
5      如果输入包含字母和下划线外的字符
6          报告词法错误，处理结束
7      将输入依据 "print" 分词
8      若得到的子串多余 1 个
9          报告语法错误，处理结束
10     依据上述分词过程得到的变量名得到要显示的变量
11 在上述情况下，若输入包含 ""
12     判断输入类型为赋值语句（将求导运算视为运算后向名为 ANS 的变量赋值）
13     如果输入包含字母和下划线外的字符
14         报告词法错误，处理结束
15     将输入依据 "" 分词
16     若得到的子串多余 1 个
17         报告语法错误，处理结束
18     依据上述分词过程得到的变量名得到要进行求导的变量，变量名设为 ANS
19     如果输入格式不是否为合法的 “变量 == 变量”
20         报告词法错误，处理结束
21     将输入依据 "==" 分词
22     若得到的子串多余 2 个
23         报告语法错误，处理结束
24     依据上述分词过程得到的变量名设置要测试相等的变量
25 在上述情况下，若输入包含 "=="
26     判断输入类型为测试相等语句
27     如果输入格式不是否为合法的 “变量 == 变量”
28         报告词法错误，处理结束
29     将输入依据 "==" 分词
30     若得到的子串多余 2 个
31         报告语法错误，处理结束
32     依据上述分词过程得到的变量名设置要测试相等的变量
33 在上述情况下，若输入包含 "=" 而不包含 "x="
34     判断输入为赋值语句

```

```

35     将输入中的 = ( ) , 替换为空格
36     如果输入包含数字, 字母, 下划线, 小数点以外的字符
37         报告词法错误, 处理结束
38     调用接收字符串的多项式构造函数
39     如果构造函数抛出异常
40         判断异常包含的信息, 设置错误类型, 处理结束
41     将得到的字符串和变量名保存在处理结果结构体中
42 在上述情况下, 若输入包含 "x="
43     判断输入为求值语句
44     使用Qt的正则表达式工具提取x的值
45     判断是否有语法错误
46 在上述情况下, 若输入由字母, +-*/()组成
47     判断输入为赋值语句 (将变量间运算视为运算后向名为ANS的变量赋值)
48     准备字符到操作符转换表, 操作符优先级表
49     将输入中的操作符字符左右各添加一个空格
50     以空格为分隔符对输入进行分词得到标识符列表
51     在标识符列表最后插入结束符 "#"
52     准备空的操作数栈和操作符栈
53     在操作符栈中压入结束符 "#"
54     遍历每一个标识符
55         若标识符是操作符
56             比较栈顶操作符与该操作符优先级大小
57             若前者小于后者
58                 在操作符栈中压入新的操作符
59             若等于
60                 操作符栈弹出一个元素
61             若大于
62                 弹出两个操作数, 一个操作符, 进行运算, 结果压入操作数栈,
63                 保持之前的操作符, 重新回到第42步判断
64 其他情况
65     报告语法错误, 处理结束

```

variable_table 映射 一个字符串类型到多项式类型的映射的变量, 在内存中储存当前变量。

statement_type 枚举 类型为 statement_type_t 的枚举变量, 储存输入的语句的类型: 赋值语句 (变量间的运算, 视为向 ANS 变量的赋值), 测试相等语句, 打印变量语句, 求值语句。

处理结果结构体

scanner_error_info, parser_error_info 用于存储词法分析和语法分析中可能出现的错误。

assign_stm_info, test_equal_stm_info, print_stm_info, eval_stm_info 储存对应输入类型的处理得到的有关信息

2.2.4 Polynomial 类

多项式中的一项使用系数与指数组成的有序对表示: $term_type = QPair < coef_type, exp_type >$ (当前实现中 $coef_type$ 和 exp_type 均为 long double)。多项式是项的线性表: $polynomial_type = QVector < term_type >$ Polynomial 的关键算法包括:

insertTerm 函数 插入新项的算法, 它有接收 QPair 以及接收系数、指数两个数字的两种重载。它插入新的项并保持各项在线性表中按照指数递减的顺序排列。算法描述如下:

- 1 判断系数、指数大小是否合法
- 2 如果不合法, 抛出异常
- 3 否则继续
- 4 判断系数是否为 0
- 5 如果为 0, 算法结束
- 6 否则继续
- 7 判断多项式中是否已经存在指数与要插入的项的指数相同的项
- 8 如果有, 抛出异常
- 9 否则继续
- 10 遍历多项式中每一项
- 11 如果该项指数小于要插入的项的指数, 跳出循环
- 12 更新记录的多项式的最大和最小的指数数值
- 13 在循环结束的项之前插入新的项

设当前项数为 N, 则算法时间复杂度为 $O(N)$

多项式加法、减法函数 首先实现函数 `operator_undertermine`, 它接收的参数包括一个二元运算符 F 分别传入 $std::plus < coef_type >$ 和 $std::minus < coef_type >$ 即可实现加法和减法, 减少了代码的重复以加法为例, 算法的描述如下:

- 1 准备空多项式 **ret**
- 2 准备迭代器 **ita**、**itb**, 分别指向相加的两个多项式的表头
- 3 当 **ita**、**itb** 不都指向表尾时
- 4 若 **itb** 指向表尾
- 5 将 ***ita** 插入 **ret**, **ita** 指向下一项

```

6      若 ita 指向表尾
7          将 *itb 插入 ret, itb 指向下一项
8      若两者均不指向对应多项式的表尾
9          若 ita 指向的项系数更高
10             将 *ita 插入 ret, ita 指向下一项
11          若 itb 指向的项系数更高
12             将 *itb 插入 ret, itb 指向下一项
13          若 ita 和 itb 指向的项系数相同
14             将两项系数相加, 并取其系数, 插入新项到 ret 之中
15             ita、itb 均指向下一项
16  返回 ret

```

设相加的多项式的系数分别为 M 和 N , 则算法时间复杂度为 $O(M+N)$

2.2.5 快速傅里叶变换多项式乘法

快速傅里叶变换法 (FFT) 可用于在 $O(n \log n)$ 的时间复杂度内完成多项式乘法。

多项式乘法问题的转化 本算法首先是把多项式相乘转换为矩阵相乘问题, 考虑多项式

$$P(x) = \sum_{i=0}^{n-1} a_i x^i$$

记其系数为向量

$$\mathbf{A} = (a_0, a_1, \dots, a_{n-1})$$

任取 n 个 x , 其中任意两个互不相同, 记

$$\mathbf{X} = (x_0, x_1, \dots, x_{n-1})$$

$$\mathbf{V}_n = \begin{bmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n-1} & \cdots & x_{n-1}^{n-1} \end{bmatrix}$$

该矩阵为范德蒙德矩阵, 当任意两个 x 不相同行列式不为 0, 存在逆矩阵将每个 x 带入 \mathbf{A} 中, 将结果记为 \mathbf{Y}

$$\mathbf{Y} = (y_0, y_1, \dots, y_{n-1})$$

有以下等式成立

$$\mathbf{V}_n * \mathbf{A}^T = \mathbf{Y}^T$$

若已知 \mathbf{X} , 则矩阵 \mathbf{V}_n 已知。若得知 \mathbf{Y} , 则 \mathbf{A} 可求

$$\mathbf{A}^T = \mathbf{Y}^T * \mathbf{V}_n^{-1}$$

接下来, 假设 $P(x)$ 是多项式 $B(x) * C(x)$ 相乘的结果

$$\begin{aligned}
 \therefore B(x) * C(x) &= P(x) \\
 \therefore Y_B * Y_C &= Y \\
 \therefore V_n * (A_B)^T &= (Y_B)^T \\
 V_n * (A_C)^T &= (Y_C)^T \\
 V_n * A^T &= Y^T \\
 \therefore A^T &= Y^T * V_n^{-1} \\
 &= Y^T * V_n^{-1} \\
 &= (Y_B * Y_C)^T * V_n^{-1}
 \end{aligned}$$

因求 $P(x)$ 就是求其系数向量 A , 故假设 $B(x)$, $C(x)$ 的系数的上限是 k , 则 $n = 2k$ 是 $P(x)$ 的系数上限。若能取 $n = 2k$ 个 x 的不同的值作为 V_n , 再求出 $Y_B * Y_C$ 即可根据式 8 求出 $P(x)$ (实现上因为使用了下文将会讲解的分治法, 因此选用的 n 是大于 $2k$ 的最小的 2 的整数幂), 有以下数学工具帮助解决这一问题

单位复数根 取 x 为 n 次单位复数根, 即 $x^n = 1$, 共有 n 个解: $x_k = w_n^k = e^{(\frac{2\pi k}{n})i}$, $k = 0, 1, \dots, n-1$, 由欧拉公式

$$e^i = \cos\theta + i * \sin\theta$$

可知, 这些 x_n 相当于在复平面上的单位圆上将单位圆平分为 n 份的坐标点 $(\cos(\frac{2\pi k}{n}), \sin(\frac{2\pi k}{n}))$, 故一定是两两不同的

单位复数根有关公式 对于单位复数根 x_n 有

$$w_{tn}^{tk} = w_t^k \quad (1)$$

$$(w_n^{k+\frac{n}{2}})^2 = (w_n^k)^2 \quad (2)$$

$$\sum_{i=0}^{n-1} (w_n^k)^i = 0, k \geq 0, k \neq n \quad (3)$$

离散傅里叶变换 取 n 个单位复数根 x_0, x_1, \dots, x_{n-1} , 带入 $P(x)$ 求得的结果向量 Y 就称为系数向量 A 的离散傅里叶变换, 其中对于每个 k (DFT)

$$y_k = \sum_{i=0}^{n-1} a_i w_n^{ki}$$

快速傅里叶变换 快速傅里叶变换利用了单位复数根的性质公式和分治法, 设

$$P_0(x) = a_0 + a_2x + a_4x^2 + \dots + a_{n-2}x^{\frac{n}{2}-1} \quad (4)$$

$$P_1(x) = a_1 + a_3x + a_5x^2 + \dots + a_{n-1}x^{\frac{n}{2}-1} \quad (5)$$

则

$$(x_n^k)^2 = x_n^{\frac{k}{2}} \quad (6)$$

$$y_0 = P_0(\mathbf{1}) = a_0 \quad (7)$$

$$y_k = P(x_k) = P(w_n^k) = P_0(w_n^{2k}) + w_n^k P_1(w_n^{2k}) \quad (8)$$

$$= P_0(w_n^{\frac{k}{2}}) + w_n^k P_1(w_n^{\frac{k}{2}}) \quad (9)$$

$$y_{k+\frac{n}{2}} = P_0(x_{k+\frac{n}{2}}) = P(w_n^{k+\frac{n}{2}}) = P_0(w_n^{2k+n}) + w_n^{k+\frac{n}{2}} P_1(w_n^{2k+n}) \quad (10)$$

$$= P_0(w_n^{2k}) - w_n^k P_1(w_n^{2k}) \quad (11)$$

$$= P_0(w_n^{\frac{k}{2}}) - w_n^k P_1(w_n^{\frac{k}{2}}) \quad (12)$$

故求出 $P_0(w_n^{\frac{k}{2}})$ 和 $P_1(w_n^{\frac{k}{2}})$ 即可求出 y_k 和 $y_{k+\frac{n}{2}}$, 对于所有 $k = 0, 1, 2, \dots, \frac{n}{2} - 1$ 求出两值即可求出所有 y_n 故每次分治问题规模减半, 根据离散数学中递归方程有关知识知整个复杂度为 $O(n \log n)$

单位复数根的范德蒙德矩阵的逆矩阵

$$V_n = \begin{bmatrix} 1 & x_0 & \cdots & x_0^{n-1} \\ 1 & x_1 & \cdots & x_1^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & \cdots & x_n^{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & w_n & \cdots & w_n^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & w_n^{n-1} & \cdots & w_n^{(n-1)*(n-1)} \end{bmatrix}$$

在求出 \mathbf{Y} 后, 只要求出 V_n^{-1} 即可求出 $P(x)$ 的系数 \mathbf{A} 有以下结论: 单位复数根的范德蒙德矩阵的逆矩阵在 (j, k) 处的值为 $\frac{w_n^{-kj}}{n}$ 故可得 $P(x)$ 的系数 \mathbf{A} 的每一项 a_i 为

$$a_i = \frac{1}{n} \sum_{k=0}^{n-1} y_k w_n^{-ki}$$

观察发现该公式与系数向量 \mathbf{A} 的傅里叶变换类似, 故在实现上可以调用同样的函数, 只要将 w_n 的次数从 ki 改为 $-ki$, 结果再除以 n 即可。

多项式长除法 本程序实现的是不带余数的多项式长除法 (余数不是多项式)

- 1 准备空多项式 **ret**
- 2 准备当前多项式的副本 **lhs** 作为被除式
- 3 当被除式的最大指数大于等于除式且被除式不为 **0** 时
 - 4 新建项 **tmp**, 指数为被除式的最大指数减去除式的最大指数
 - 5 系数为被除式的最大指数项的系数除以除式的最大指数项的系数
 - 6 在被除式中减去 **tmp*除式**
 - 7 在 **ret** 中插入 **tmp**
- 8 返回 **ret**

3 程序使用和测试说明

3.1 程序使用

3.1.1 安装说明

本程序有 Windows 和 Mac 两个版本，安装方法有所不同

- Windows 版本无需安装，解压压缩文件在 release 文件夹中找到 calc.exe，打开即可使用。
- Mac 版本打开装载本程序的 calc.dmg 文件，将本程序拖入 Application 文件夹即可完成安装，在 Launchpad 中可以找到。至少运行在 MacOS 10.12 下。

3.1.2 使用说明

基本用法是 用鼠标点击下图中红色箭头所指向的主输入条以开始输入，按下输入按钮或回车键输入命令



支持的语句包括

- 输入多项式，如 $p = (1,2)(3,4)(2,0)$ 。等号左侧是变量名，支持下划线和字母。右侧是多项式，使用 (系数 1, 指数 1)(系数 2, 指数 2) ... (系数 n, 指数 n) 的形式表达。输入正确会在主输出窗口中显示指数从大到小排列的多项式。在变量窗口中会显示该变量。

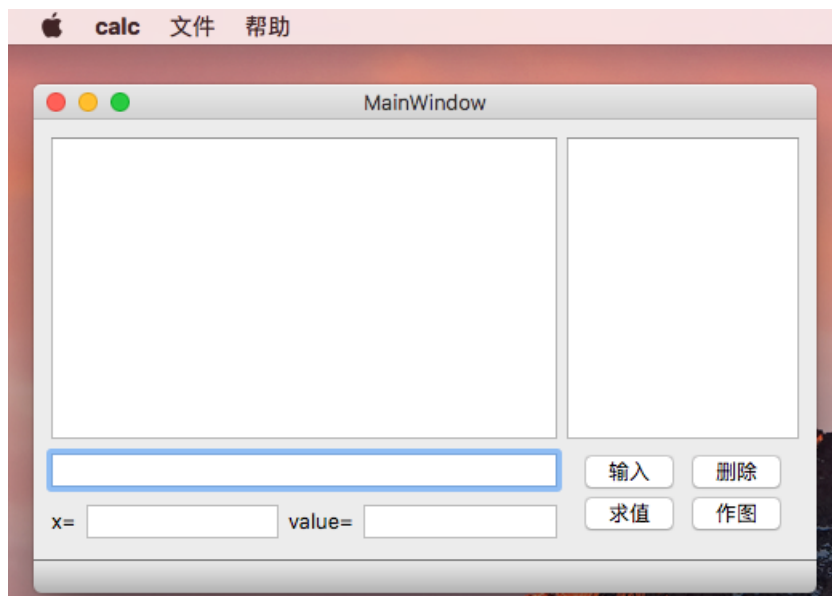
- 输出多项式，如 `print p` 或者 `print all`，输出结果显示在主输出窗口。
- 多项式运算，如 $p+q$, $p-q$, $p*q$, $p==q$, p' , $p[x=3]$, $p*(q+r)-s$ 。其中加减乘除和括号支持混合运算。其他运算符不支持。运算结果会输出在主输出窗口。

其他用法包括

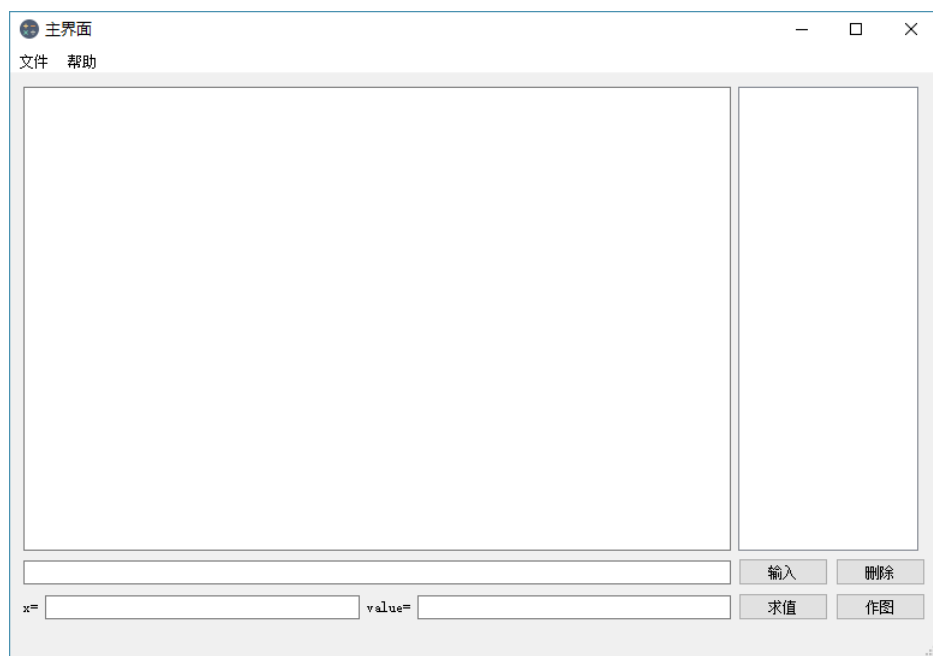
- 在主输入条输入合法的多项式或会得到一个多项式结果的多项式运算后，在 x 输入条输入 x 的值，点击求值或按下回车将会在求值输出条和主输出窗口中显示求值结果，主输入的求值语句不影响求值窗口结果。
- 双击变量窗口中的一个变量可以将其显示在主输出窗口中，选中后点删除可删除这一变量。
- 在主输入条输入合法的多项式或会得到一个多项式结果的多项式运算后，点击作图，将打开作图窗口，输入合法的 x 范围后可作出该多项式函数的图像。
- 在点击菜单栏文件菜单，可以将当前运算结果和变量保存到文本，或从之前保存的文本中加载运算结果和变量。

跨平台说明

- 程序界面在不同操作系统或编译环境下可能不同，特别是：Mac 系统下与 Windows 系统下菜单栏位置不同，Mac 系统下菜单栏在系统栏最上方，而 Windows 系统的在窗口内



Mac 下的界面



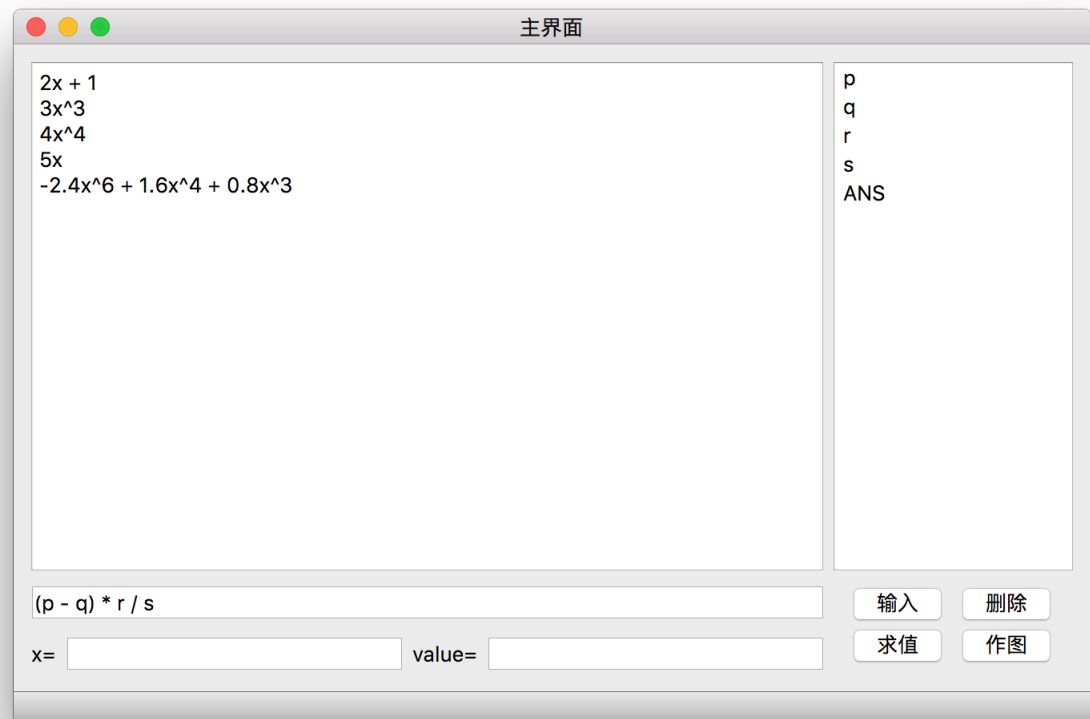
Windows 下的界面

- 本程序中允许的指数和系数的上下界的定义是 `double`, `int` 类型的最大值和最小值（有符号的），与具体机器实现有关

菜单栏对比

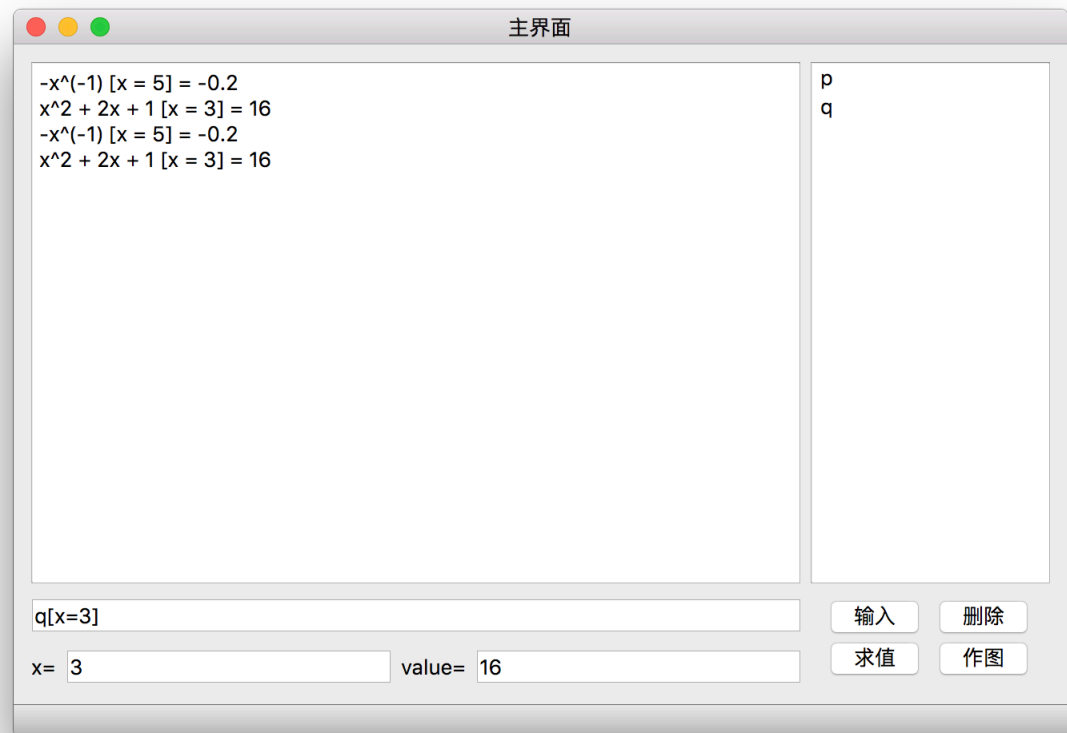
3.2 四则运算测试

- $p = (1, 0)(2, 1)$
- $q = (3, 3)$
- $r = (4, 4)$
- $s = (5, 1)$
- $(p - q) * r / s = -2.4x^6 + 1.6x^4 + 0.8x^3$



3.3 求值测试

- 通过点击求值按钮求值
- $p = (-1, -1), x = 5$
- 求得 $-x(-1)[x = 5] = -0.2$
- $p = (2, 2)(3, 1)(1, -1)$
- 求得 $x^2 + 2x + 1[x = 3] = 16$
- 通过命令求值
- $p[x = 5]$
- $q[x = 3]$



3.4 求导测试

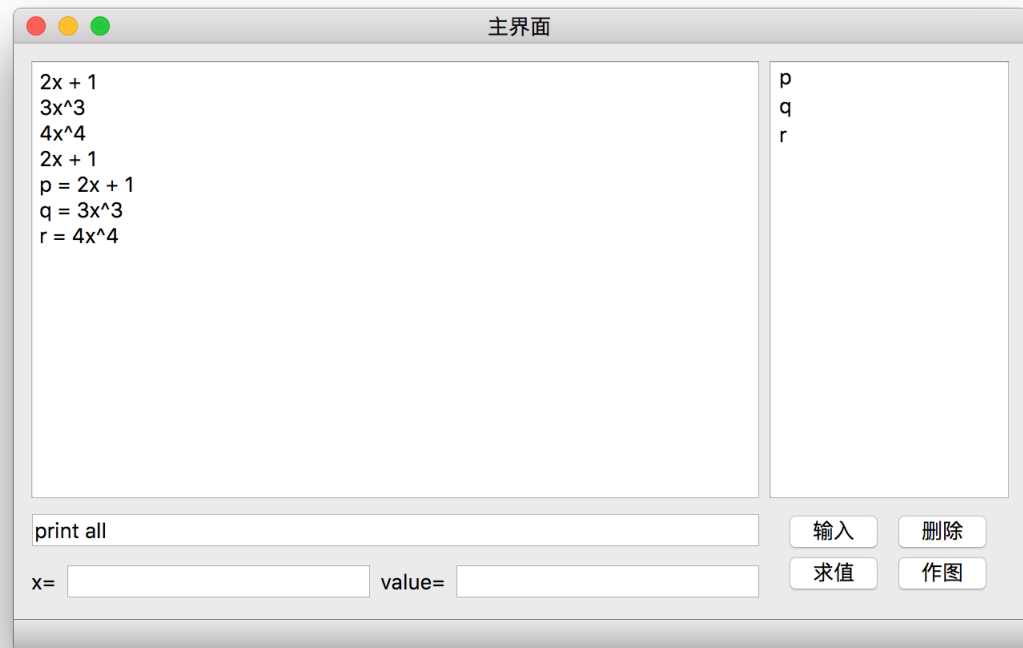
- $p = (2, 2)(2, 1)(1, 0)$
- 求得 $p' = 4x + 3 - x(-2)$



3.5 打印变量测试

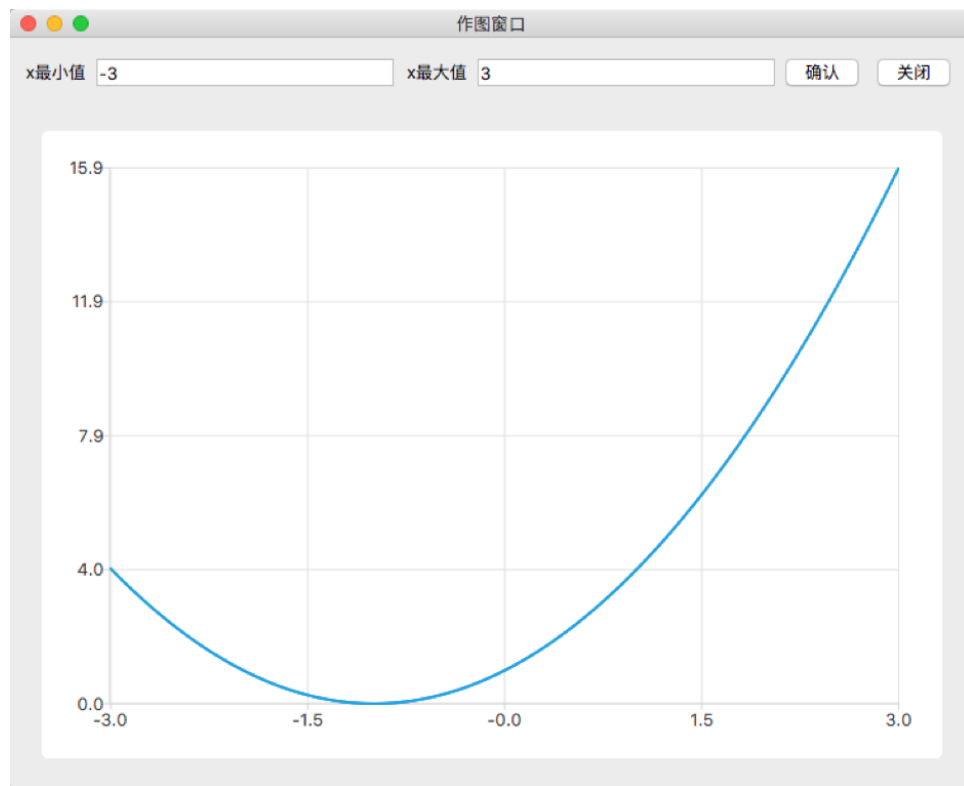
- 输入:
- $p = (1, 0)(2, 1)$
- $q = (3, 3)$
- $r = (4, 4)$
- print p
- print all
- 输出:
- $2x + 1$
- $p = 2x + 1$
- $q = 3x^3$

- $r = 4x^4$

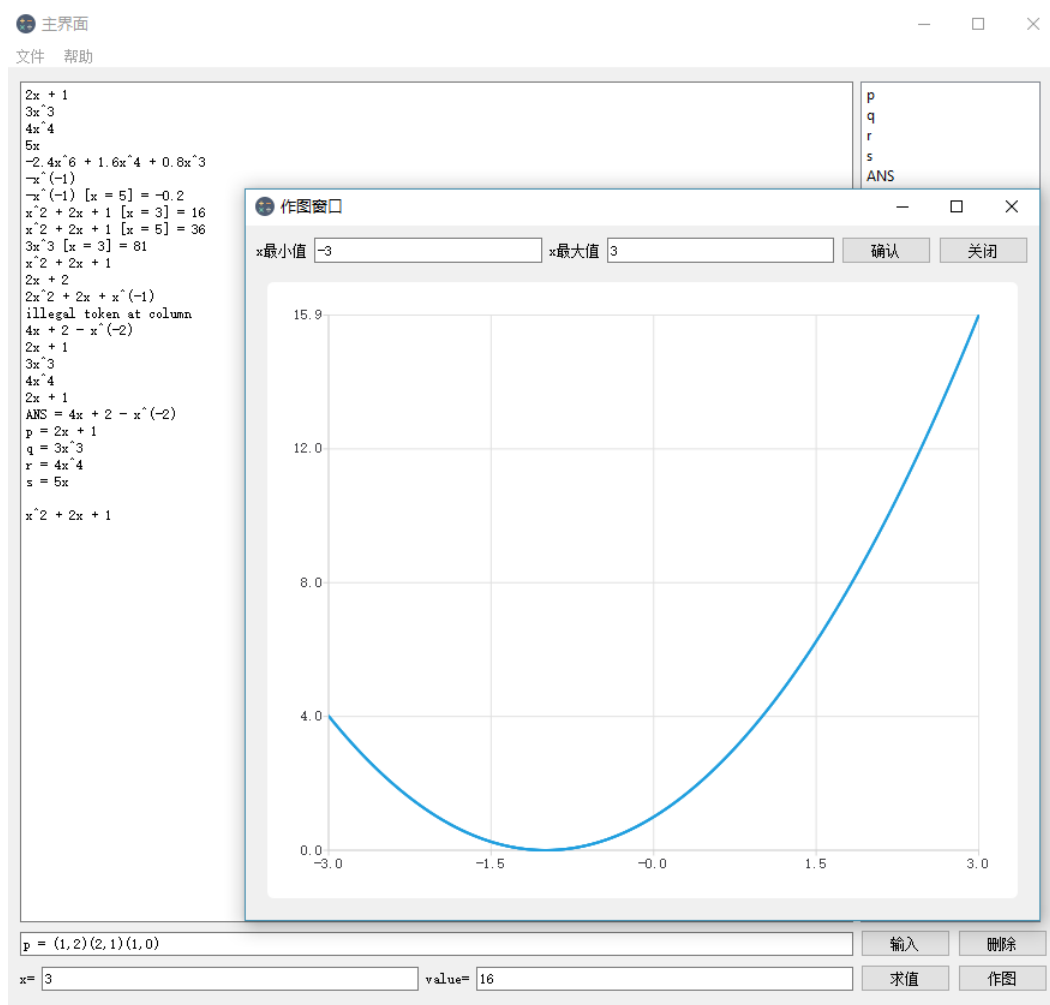


3.6 作图测试

$x^2 + 2x + 1$ 在 $[-3, 3]$ 范围内



以上测试全部在 Windows 平台上同样进行过



3.7 错误提示

本程序基于通过上文提到的三个记录错误信息的结构体和 C++ 的异常机制，通过反复测试各种输入，目前能够处理各种错误而不崩溃，并给出提示。包括：

- 非法字符
- 部分的语法错误
- 引用未定义的变量
- 输入数值超出计算能力
- 除以 0
- 读写文件错误

- 作图 x 范围错误。

4 总结讨论

通过国庆节 8 天及开学 4 天来的学习和编程，我成功完成了该计算器项目。

我收获的是

- 学会了 FFT 算法的原理
- 通过实现多项式库和基于栈的用户输入处理函数增强了实现数据结构和算法的能力
- 通过撰写多个分级模块的系统提高了工程能力
- 提升了使用图形界面编程的熟练度
- 提升了使用 \LaTeX 撰写文档的熟练度

特色是

- 使用图形界面，支持绘图
- 支持保存运算结果到文件和从文件中读取

还可以改进的地方是

- 本程序的错误处理还不是很完善，对于语法错误和此法错误还不能提示出具体的原因。我了解到，计算器处理字符串输入的原理和编译器前端处理源代码的方式别无二致，期待在将来的版本中，使用正则表达式转 DFA 算法进行词法分析，使用 LL 分析法进行语法分析，使得我的计算器能够应对更多样，更类似自然的数学语言的输入。

5 致谢和参考文献

- 感谢王若梅老师在数据结构课程上的教导，本程序的基于栈的四则运算算法的实现是基于她课堂上关于该算法的描述。
- 感谢某博客用户 @397915842 撰写的细致的 FFT 算法解析 <https://www.zybuluo.com/397915842/note/37965>。这篇文章让我学会了基于 FFT 的多项式乘法的原理，从而能实现出该算法。
- 感谢 Qt 开源项目 <https://github.com/qt> 和 Qt 官方文档库 <https://doc.qt.io/>，没有这个项目和它完整的文档支持，我无法实现本程序的图形界面功能。
- 感谢图标网站 <https://www.iconfinder.com> 为我提供了本程序的图标。