

# Minimal BASIC 设计文档

## 1. 组成

- 1) **console 类**: 完成文本框的绘制、文本的输出、输入功能
- 2) **expression 类**: 处理表达式, 完成中缀转后缀以及表达式树的建立
- 3) **line 类**: 用于按行号储存每行代码
- 4) **mainwindow 类**: 主窗口的显示, 各种类对象的实现, 代码的读取、运行等。
- 5) **statement 类**: 不同语句的执行
- 6) **main 函数**

## 2. 功能及其实现

1) **console 类**: 主要是利用 QBoggle 中助教提供的 console 类的代码写成。继承自 QTextEdit, 通过监控键盘事件读取输入, 调用 QTextEdit 中的 append() 函数完成一行字符串的输出显示。

2) **expression 类**: 父类 Expression 以及三个子类 constantExp, identifierExp, compoundExp. 父类中有一个中缀转后缀的函数, 用于把用户输入的表达式转化为后缀表达式并产生一个 QList, 父类的构造函数包括表达式的字符串和整个代码共享的变量名与其对应值的 QMap varlist. constantExp 用于常量, 将一个字符串的数字转化为对应的十进制数储存起来, 可以通过函数 value() 调用这个常数值。identifierExp 用于变量, 利用 varlist 可以把该变量的值查出或添加新变量。compoundExp 用于复合表达式, 通过递归求值。

3) **line 类**: 用于将一个字符串, 即用户输入的代码, 分为行号和实际的代码对应储存到 QMap 中。供 mainwindow 调用。

4) **statement 类**: 父类为 statement, 构造函数为表达式的字符串和变量列表。子类为各种语句的对应 statement 类: letstmt, printstmt, inputstmt, gotostmt, ifstmt. 分别利用其 run() 函数和 expression 类完成对应功能。letstmt 类操作赋值语句, 通过 "=" 符号将表达式分为左值和右值, 左值为一个变量, 右值为 compoundExp, 将左值对应右值存入变量列表。printstmt 读取要打印的变量名并返回其值。inputstmt 仅用于返回要求输入的变量名, 输入的主要操作在 mainwindow 中完成。gotostmt 返回表达式的对应的行号的数值。ifstmt 通过 THEN 判断条件部分和操作部分。判断条件部分根据 > < = 号判断左右两端 compoundExp 的值是否满足条件, 满足条件则将 gotonum 赋值为操作部分的行号的数值, 否则 gotonum 默认值将为 -1, 即无对应行。

5) **mainwindow 类**: 有着上述各类的实现和主窗口的显示。

通过槽函数 addNewLine 与 console 的信号连接, 处理每一次新输入的代码内容。通过 line 类的指针 codes 储存每一行的内容。

通过 runCmd() 函数判断用户输入指令并进行对应操作

run() 函数通过依次调用每一行代码内容并利用 statement 类进行操作

help(), list(), clear(), quit() 分别进行对应指令的操作。

比较特殊的是对应执行 input 语句的功能, 添加了行号标记的变量 stopNum 和新的槽函数 iptValue(), 当需要输入时记录已执行到的行号, 断开原有信号槽的连接, 结束 run

( ) 函数, 连接新的信号槽, 读取输入内容, 格式无误方储存到对应的变量列表中。然后根据之前停止的行号继续开始 run() 函数。

### 3. 附加功能

- 1) 附加了函数功能。具体实现见代码及注释
- 2) 附加了\*\*运算符。具体实现见代码及注释