

Project2 说明

1. 原理简述

本 project 是用 Qt 实现的简单的 Basic 解释器。主要是对不同 statement 进行不同处理，并把 expression 化成表达式树，然后前序遍历表达式树进行计算。

2. 窗口组成部分说明

左上角的窗口是 Code console，用来输入相关的 Basic 代码

左下角的窗口是读到 INPUT 命令后，打出“？”，使用者可以在后边输入变量的值

右上角的窗口显示变量列表

右下角的窗口用来显示 HELP，报错等各种提示

3. 操作说明

Project 实现了 PDF 中各种要求，拓展功能实现了 DEBUG 功能。通过直接输入 DEBUG 可以进入 debug 模式。输入 B linenum 可以设置断点，例如 B 110 是在第 110 行设置断点。输入 CONTINUE 可以一直运行到下一个断点。输入 STEP 可以进行单步调试。

4. 各个类的介绍

- (1) console 是用来构造各个输入区的类，基本复制了 QBoogle 里边的 console，当输入回车时即认为输入一条新的命令。
- (2) evaluate 是一个包装过的 map，存储了已经定义的变量，并支持输入新的变量，也可以通过输入变量的 name 获得变量的 value。
- (3) statement 存储各条指令。
- (4) expression 存储表达式，同时可以计算 expression 的值
- (5) program 是一个链表，存储带行号的 statement。
- (6) tokenizer 是一个分割器，可以把 expression 按+*/=分割成不同的部分。
- (7) parser 则将已经分割好的表达式建立成表达式树
- (8) mainwindow 是最顶层，将用户输入的各种命令分类处理

5. 目前存在的一些问题

(1) 因为偷懒直接用了 Console 显示了提示和变量表，也就是说用户可以对这两个区域进行修改(尽管不会对程序影响产生影响)

(2) 因为 Console 是使用回车键作为一条 statement 的结束，并进行存储的。这就造成了一旦按下回车键，就不能再直接在 console 里进行修改的情况，必须再写一行进行覆盖。。如果强行修改的话也不会进行存储，会报错或者导致运行错误。

(3) 程序的架构问题很大，理论上讲 RUN 之类的函数都应该在 statement 里实现，但是图方便都放到了 mainwindow 里，导致 mainwindow 里代码量爆炸，也不好修改。

(4) 后期修改的时候，为了图方便，都是加代码，而不是对原来的代码进行修改。如重写了个 ConditionRun，其实都是复制了 RUN，完全可以写成一个函数。