**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。

1. **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

1. **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。

1. **各个类的介绍**
2. 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。
3. 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。
4. statement存储各条指令。
5. expression存储表达式，同时可以计算expression的值
6. program是一个链表，存储带行号的statement。
7. tokenizer是一个分割器，可以把expression按+-\*/=分割成不同的部分。
8. parser则将已经分割好的表达式建立成表达式树
9. mainwindow是最顶层，将用户输入的各种命令分类处理
10. **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。