**编译原理第一次程序实验报告**

**实验目的**

以一个字符串的形式读入一个文件内容，然后输出符号对应的类型和种别码

**完成的模块和模块流程**

扫描的过程参考了《自制编译器》

1. 运行环境：

运行在JDK 10上 使用了Kotlin编写的代码，调试环境是Jet Brains Intelli J

1. 完成内容

从一个文件中读入一串代码进行分析，然后通过先前已经规定的种别码和字符的对应表进行选择，最后输出放入到一个返回的数组中。

1. 实验流程

打开输入流，读取对应路径下面的文件中的字符代码，通过Java split函数进行简单分段，最后再顺序扫描进行处理，最后将token和种别码分别放到list 中输出。

使用多条测试用例，均可以通过实验

1. 实验代码的迭代：

第一代的代码使用了顺序扫描的办法

第二代的代码使用和应用了正则表达式处理复杂的表达式内容，比如：

X = y = x + 1

第三代的放弃了正则表达式处理，并且对一些算法和逻辑进行了一些优化并且支持中文变量名称命名。

**关键技术和难点**

1. 如何区分一个关键字和表达式？
2. 能否调用递归程序解析代码？
3. 能否使用正则表达式处理？
4. 错误/异常处理：

主要思路是先使用Java 中的字符串处理函数先处理一遍整个程序，将程序中的表达式分割成关键字和符号还有表达式，然后在对表达式分解。 关键字和符号储存在一个map中包括他的种别码，这样分解起来比较愉悦。

Q：如何区分一个关键字和表达式

A：关键字定义在一个Table.kt文件中，通过扫描token串可以得出这个token是否在关键字，如果不再定义的Table中则放入后面的代码中扫描，依次判断是否是数字 变量名称 表达式等等，如果都不是，最后将加入error 到输出中。

变量命名的支持：

1. 支持以非数字开头的变量名称(可以以下划线开头)

[\_]\*[a-z|A-Z|\u4e00-\u9fa5][0-9|a-z|A-Z]\*

配置这样的正则表达式

Q：能否使用递归程序处理代码：

A：递归程序主要想使用在处理复杂的表达式中，比如 x := x := y的情况，但是开始的时候使用了对空格进行分串的方法，所以这样的使用递归程序不方便。

Q：能否使用正则表达式处理：

A:因为对token空格进行了分串，使用正则表达式美发正常扫描，就算可以使用正则表达式，直接使用顺序遍历也可以完成...

([a-z][a-z|0-9]\*)([-+\*/:=][-+\*/]\*=\*)(.\*)

但是这样的表达式在表达式中没有空格的时候无法运算。

像这样的正则表达式可以扫描出上面的没有空格的例子，但是有了空格之后没法正常扫描。

**编程调试程序的难点**

1. 正则表达式调试问题：

开始的时候正则表达式调试还是很难受的，大概熟悉了语法之后就方便很多。后来也加入了中文表达式命名等

1. 字符转换问题：使用的编程环境是IDEA 使用还是非常好的，使用的语言是Kotlin，这是一种完全面向对象的语言，没有Java 中的基本类型 如char 类型 是Char 类，其中地toInt() 函数是直接将字符直接转化为对应的ascii 码中地数字，需要减去 48 才是真实地数字，开始的时候没有考虑这一点，字符转换的

3.运算符处理的时候有时候会漏掉一些case，需要反复修正才可以