



华南师范大学

## 本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：编译原理课程项目

实验项目：中缀表达式转后缀表达式

指导老师：王欣明

开课时间：2018 ~ 2019 年度第 2 学期

专 业：计算机科学与技术

班 级：网工 7 班

学 生：林瑞儿+曹博凯+卓伟洲

学 号：20162180014+20162382018+20162180063

华南师范大学教务处

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063  
专 业 网络工程 年级、班级 网工 7 班  
课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式  
实验时间 2019 年 6 月  
实验指导老师 王欣明 实验评分

## 一、实验内容

中缀表达式转后缀表达式

## 二、实验要求

- 1、实现一个完整的 Java 程序，它读取文件中的中缀表达式（每个表达式以分号结束，文件中可以有多个表达式）并转换为等价的后缀表达式后输出到屏幕上。
- 2、表达式中的运算量可以是任意整数或者小数，支持加、减、乘、除、取负运算以及小括号，表达式中的空格、制表符等空白符号可以被忽略。
- 3、若用户输入的表达式有误，则提示用户错误的位置。譬如两个运算量之间缺少运算符、或运算符缺少左（或右）运算量等。错误处理功能的最低要求是当输入表达式有错时，给出一个报错信息，提示错误的位置和类别。  
学有余力的学生还可考虑尝试如何实现出错恢复（Error Recovery），即当程序发现一个错误时不是立马停下来，而是能够从跌倒的地方爬起来，继续分析下去，从而一次运行即可发现更多的错误。
- 4、为以上的 Java 程序提供一个随机测试数据发生器（用 Java 语言来写），生成若干随机的正确表达式和不正确表达式（通过命令行参数决定是生成正确的还是不正确的以及生成的数量）。生成的测试数据要求写入文件，可以被 1 中的程序读取。

## 三、运行环境

- 1、操作系统：Windows10
- 2、软件：eclipse
- 3、硬件：i5-520m CPU+8G 内存

## 四、实验原理

在计算一个表达式的时候，可以用数据结构中栈的知识，将我们平常熟悉的中缀表达式转为后缀表达式，再将后缀表达式进行计算得到结果。先说下什么是中缀什么是后缀：

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学号 20162180014 20162382018 20162180063  
专 业 网络工程 年级、班级 网工 7 班  
课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式  
实验时间 2019 年 6 月  
实验指导老师 王欣明 实验评分

中缀表达式: eg:  $9 + (3 - 1) * 3 + 10 / 2$ , 就是我们平常计算时的表达式;

后缀表达式: eg:  $9\ 3\ 1 - 3 * + 10\ 2 / +$ , 不包含括号, 运算符在两个运算对象后面的表达式。

## 1. 中缀表达式转后缀表达式

我们先初始化一个空的字符串(String)和栈(Stack), 将这个中缀表达式中的数字和运算符(包括加减乘除及左右括号)分割成一个个字符串存进一个新的数组中, 比如上面的中缀表达式就存为:

9 + ( 3 - 1 ) \* 3 + 10 / 2 (简便起见用空格将一个个字符串隔开了...) 然后遍历这个数组, 运算对象就直接放进刚才初始化的 String 中, 栈中存放的是运算符。如果是空栈, 加减乘除或左括号就直接进栈, 如果栈不为空, 此时扫描到的运算符与栈顶运算符做优先级比较, 如果栈顶运算符优先级低于扫描的运算符, 则当前扫描运算符入栈, 否则栈顶运算符弹栈并连接在 String 后, 并与弹栈后的站定运算符再做比较, 直到站定运算符低于当前扫描的运算符或遇到左括号, 并将当前运算符进栈; 当前运算符若是右括号, 则将栈顶运算符依次弹出并依次连接到 String 后, 直到遇到左括号, 并将左括号弹出(但是不连接在 String 后的, 注意后缀表达式中是没有括号的)。如果中缀表达式遍历完毕后栈中还有运算符则将栈中剩下的运算符依次弹出并连接在 String 后面; 最终得到的就是字符串 String 后缀表达式。

## 2. 后缀表达式的计算

先初始化一个栈, 这个栈是用来存放运算对象的, 然后定义一个变量存放最后的结果。遍历到数字则依次进栈, 遍历到运算符的时候将栈顶运算符弹出(假设赋给变量 a), 再将此时的栈顶运算符弹出(假设赋给变量 b), 用后者对前者做该运算符对应的运算(假设遍历当前的运算符为+, 则做计算  $b+a$ ), 然后将计算结果入栈。以此方式遍历整个字符串, 最终的结果即为运算结果。

例如:

中缀表达式 “ $9 + (3 - 1) * 3 + 10 / 2$ ” 通过上述步骤后转化为后缀表达式的结果便是: “ $9\ 3\ 1 - 3 * + 10\ 2 / +$ ”

## 五、具体实验设计

### 1、中缀转后缀

(1) 分号入栈, 从左到右遍历中缀表达式的每个数字和符号

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063  
专 业 网络工程 年级、班级 网工 7 班  
课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式  
实验时间 2019 年 6 月  
实验指导老师 王欣明 实验评分 \_\_\_\_\_

- (2) 若为数字就输出, 即成为后缀表达式的一部分
- (3) 若为加减乘除, 则判断其与栈顶符号的优先级, 优先级低于栈顶符号 (乘除优先加减) 则栈顶元素依次出栈并输出, 并将当前符号进栈
- (4) 若为左括号, 进栈
- (5) 若为右括号, 需要去匹配此前的 “(”, 所以栈顶依次出栈, 并输出, 直到 “(” 出栈为止
- (6) 扫描到分号时, 将栈中符号全部出栈并输出。

另外:

核心代码:

下面是实现的一个表达式分析器的类:

```
class StateAnalyze {  
  
    public LinkedList<Toke> tokes;  
    public Toke tok;  
    public int index;  
    String fail;  
    public String output="";  
  
    public StateAnalyze(LinkedList<Toke> list)  
    {  
        this.tokes=list;  
    }  
    public void Error(String a) throws CodeException  
    {  
        System.err.print(a);  
        throw new CodeException(a);  
    }  
    public void match(String a) throws CodeException  
    {  
        if (tok.word.equals(a) || tok.type.equals(a))  
        {  
            if (index<tokes.size())  
            {  
                tok=tokes.get(index++);  
            }  
        }  
    }  
}
```

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063

专 业 网络工程 年级、班级 网工 7 班

课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式

实验时间 2019 年 6 月

实验指导老师 王欣明 实验评分

```
    }
    else
    {
        Error("wrong word! expected: "+a+
actually:"+tok.word+"index:"+ (index+1) +";line:"+tok.line);

    }
}

public void start_analyse()
{
    index=0;
    try {
        tok= tokens.get(index++);
        exp();
    } catch (CodeException e)
    {
        fail=e.message;
    }
}

public void exp() throws CodeException
{
    term();
    while(tok.word.equals("+") || tok.word.equals("-"))
    {
        String s=tok.word;
        match(tok.word);
        term();
        output+=s;
        output+="|";
    }
}

public void term() throws CodeException
{
    Negative();
    while (tok.word.equals("*") || tok.word.equals("/"))
    {
        String s=tok.word;
```

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063

专 业 网络工程 年级、班级 网工 7 班

课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式

实验时间 2019 年 6 月

实验指导老师 王欣明 实验评分

```
        match(tok.word);
        Negative();
        output+=s;
        output+="|";
    }

}

public void Negative() throws CodeException
{
    if (tok.word.equals("-"))
    {
        String s=tok.word;
        match("-");
        output+=s;
        factor();
    }
    else factor();
}

public void factor() throws CodeException
{
    char c=tok.word.charAt(0);
    if (tok.word.equals("("))
    {
        match("(");
        exp();
        match(")");
    }
    else if (tok.type.equals("number"))
    {
        int temp=Integer.parseInt(tok.word);
        String s=tok.word;
        match("number");
        output+=s;
        output+="|";
    }
    else Error("不是合法的表达式!!! (列:"+index)+", 行:"+tok.line+"");
}
```

# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063

专 业 网络工程 年级、班级 网工 7 班

课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式

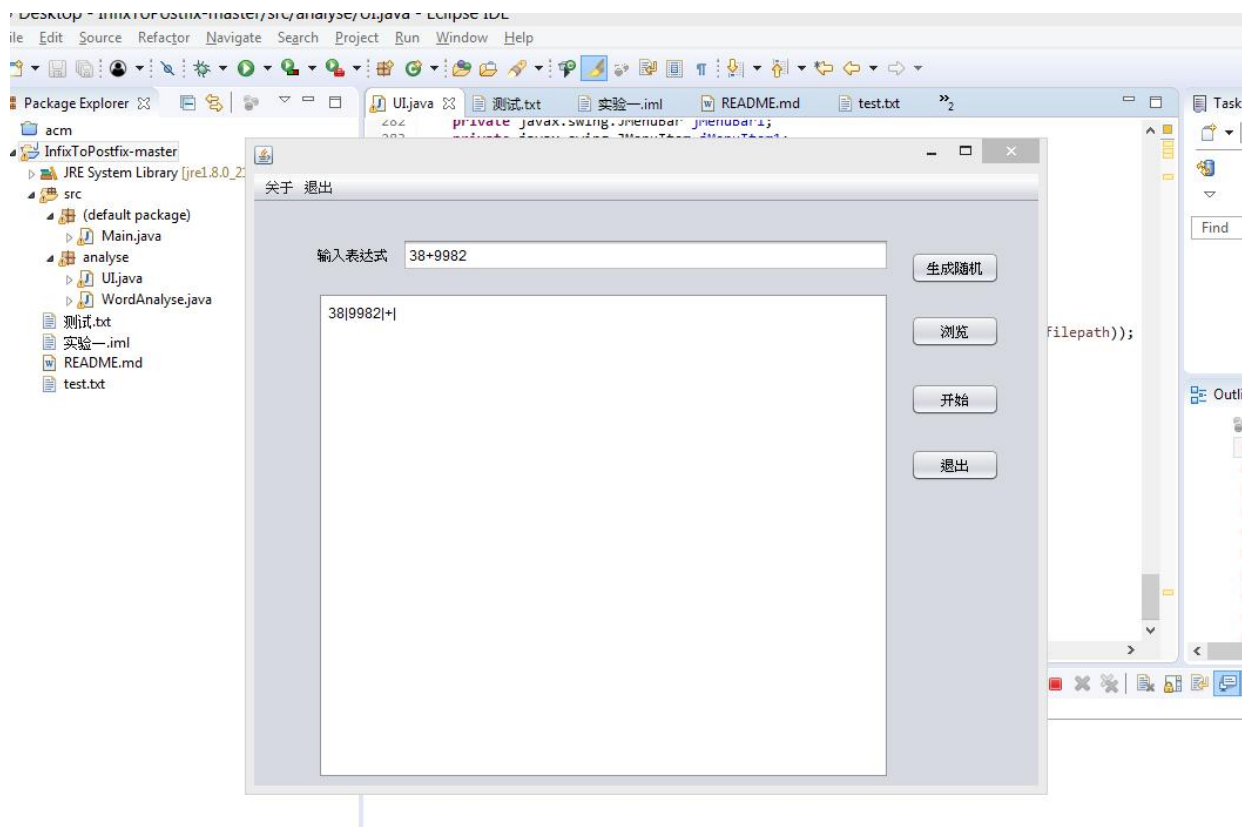
实验时间 2019 年 6 月

实验指导老师 王欣明 实验评分

```
public static void main(String args[])
{
    Users users=new Users("test.txt");
    System.out.println(users.wordAnalyse.tokes);
    users.stateAnalyse.start_analyse();
    System.out.println(users.stateAnalyse.output);
}
```

完整源代码见 中缀转后缀文件夹

## 六、运行结果：如图所示



# 华南师范大学实验报告

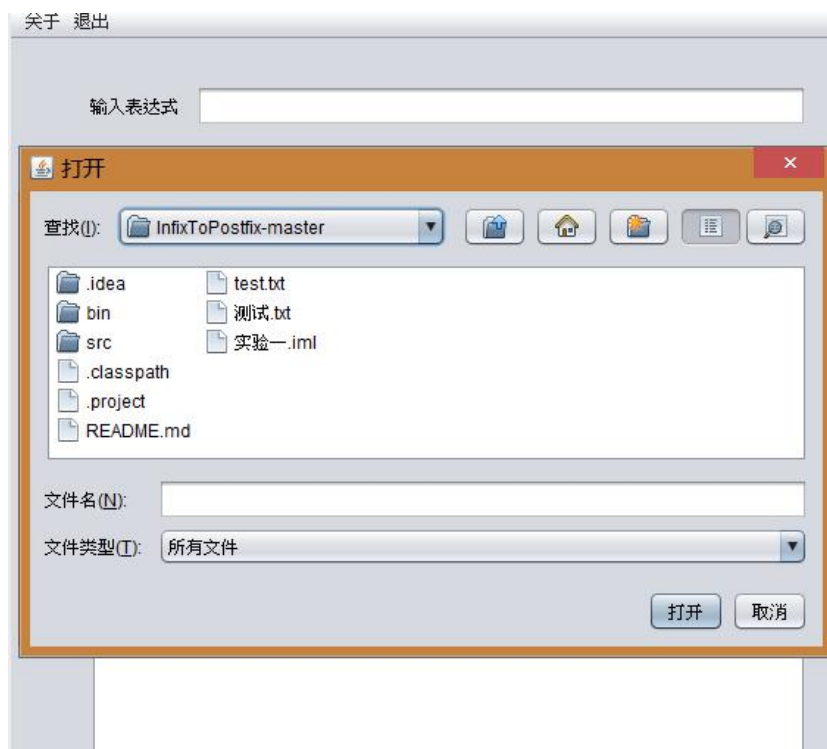
学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063

专 业 网络工程 年级、班级 网工 7 班

课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式

实验时间 2019 年 6 月

实验指导老师 王欣明 实验评分





# 华南师范大学实验报告

学生姓名 林瑞儿, 曹博凯, 卓伟洲 学 号 20162180014 20162382018 20162180063  
专 业 网络工程 年级、班级 网工 7 班  
课程名称 编译原理课程项目 实验项目 中缀表达式转后缀表达式  
实验时间 2019 年 6 月  
实验指导老师 王欣明 实验评分



## 七、实验心得与体会

- 1、通过实验，进一步理解了中缀表达式转后缀表达式的原理以及具体代码实现过程，加强了编程能力以及使用数据结构解决一些实际问题的能力。
- 2、通过团队协作，共同完成实验作业，提高了沟通交流合作的能力，在实验过程中遇到过不少困难，最终通过交流合作克服困难解决问题，收获良多。