

学生学号	0122010870125	实验课成绩	
------	---------------	-------	--

武汉理工大学

学生实验报告书

实验课程名称	编译原理
开 课 学 院	计算机与人工智能学院
指导教师姓名	王云华
学 生 姓 名	雷裕庭
学生专业班级	软件 sy2001 班

2022 -- 2023 学年 第 一 学期

实验课程名称：编译原理

实验项目名称	词法分析程序的设计与实现			实验成绩	
实验者	雷裕庭	专业班级	软件 sy2001	组别	词法解譯のフォーリズム
同组者	雷裕庭 熊壮 张艺晨 程英豪			实验日期	2022 年 11 月 24 日

第一部分：实验分析与设计（可加页）

一、实验内容描述（问题域描述）

【问题描述】DFA 化简问题的一种描述是：

请根据给定的文法设计并实现词法分析程序，从源程序中识别出单词，记录其单词类别和单词值，输入输出及处理要求如下：

（1）数据结构和与语法分析程序的接口请自行定义；类别码需按下表格式统一定义；

（2）为了方便进行自动评测，输入的被编译源文件统一命名为 testfile.txt；

输出的结果文件统一命名为 output.txt，

结果文件中每行按如下方式组织：

单词类别码 单词的字符/字符串形式(中间仅用一个空格间隔)

【类别码标准】

单词名称	类别码	单词名称	类别码	单词名称	类别码	单词名称	类别码
标识符	IDENFR	if	IFTK	-	MINU	=	ASSIGN
整型常量	INTCON	else	ELSETK	*	MULT	;	SEMICN
字符常量	CHARCON	do	DOTK	/	DIV	,	COMMA
字符串	STRCON	while	WHILETK	<	LSS	(LPARENT
const	CONSTTK	for	FORTK	<=	LEQ)	RPARENT
int	INTTK	scanf	SCANFTK	>	GRE	[LBRACK

char	CHARTK	printf	PRINTFTK	>=	GEQ]	RBRACK
void	VOIDTK	return	RETURNTK	==	EQL	{	LBRACE
main	MAINTK	+	PLUS	!=	NEQ	}	RBRACE

二、实验基本原理与设计（包括实验方案设计，实验手段的确定，试验步骤等，用硬件逻辑或者算法描述）

【实验原理】

词法分析包括：用户自定义标识符、常数、字符串、关键字、界符的识别。用户自定义符号，顾名思义就是自己定义的变量名；函数名，常数包括整数、浮点数、科学计数；字符串包括 ‘ ’ 、 “ ” 两种形式的字符串；关键字就是程序内置的关键字，如 int、main 等；界符就是各类符号，如运算符、{}、[]等。

词法分析的任务是，给定输入，识别输入序列中的单词，并将其正确分类。对于重复出现的单词，应该保证唯一性。

词法分析说容易也容易，人一眼就可以看出来一个”词“是属于哪一类的，然而让计算机来做就需要抽象出规则。常用的词法分析处理思路就是使用 DFA，这在常数识别上体现得淋漓尽致。

这个词法分析器是一个“总分结构”，各种识别功能在子程序写好，然后在主控程序里调用即可。解释一下入口条件和出口条件：入口条件是主控程序的判断条件，指针读到的第一个字符决定主程序应该调用哪个子程序。出口条件是子程序执行期间的判断条件，决定何时单词识别结束、可以跳出。

【求解步骤】

1. 定义语言子集并编码，要求有保留字、标识符、常数、运算符、界符。
2. 画出状态转换图。
3. 编写代码，实现分析。

三、主要仪器设备及耗材

PC 机

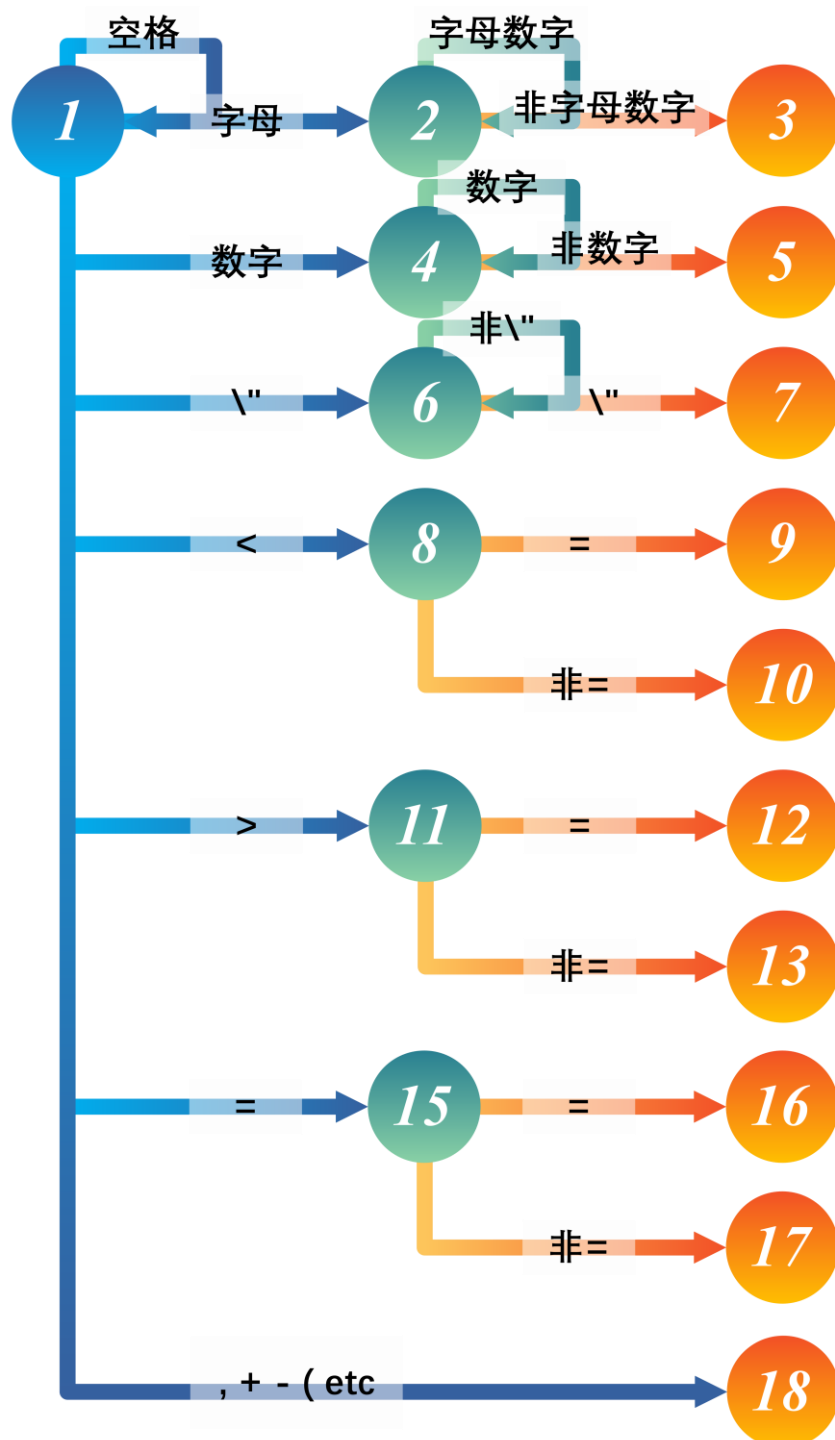
PyCharm 编译器

第二部分：实验调试与结果分析（可加页）

一、调试过程（包括调试方法描述、实验数据记录，实验现象记录，实验过程发现的问题等）

下面是主要的程序设计说明

1. 基于实验设计的状态转换图如下



软件sy2001

雷裕庭 绘制

识别实验五语言的状态转换图

2.主要的函数设计和调用

```
{'name': '标识符', 'class': 'IDENFR'},  
{'name': '字符常量', 'class': 'CHARCON'},  
{'name': '字符串', 'class': 'STRCON'},
```

strClean:

该函数处理以下问题

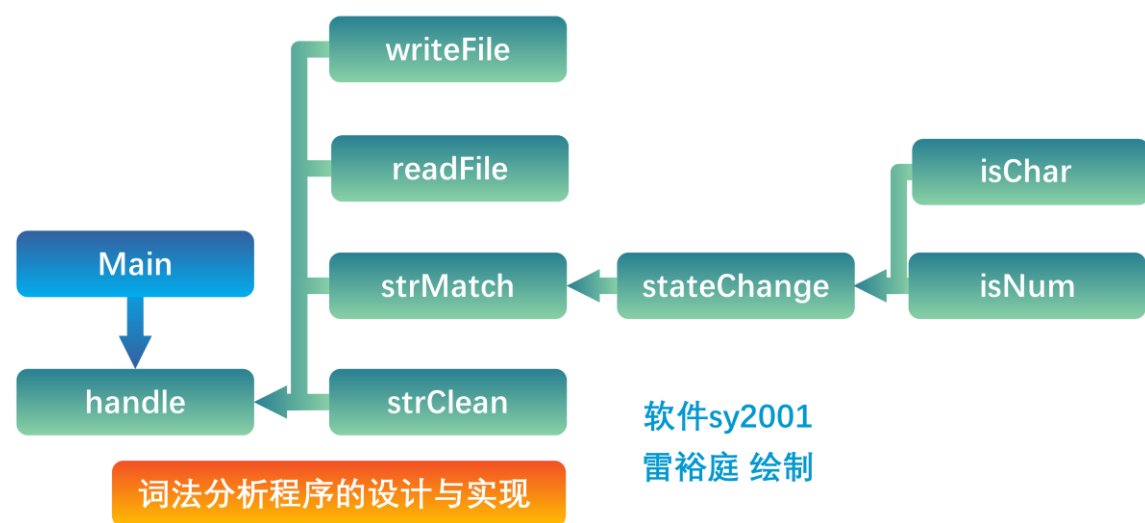
- 1.将 STRCON 从 CHARCON 中分出来
- 2.将 IDENFR 从 CHARCON 中分出来

strMatch:

不断的读取字符串的字符，调用 **stateChange** 函数

stateChange:

用于基于上一步状态和下一步的输入 char，判断下一步的状态和进行状态的储存



3. 主要实验代码

'''

实验五 11.21 软件 sy2001 雷裕庭开始编写
11.24 编写完成

'''

'''

isChar 判断是否为字符

'''

```
def isChar(input):
```

```
    input=str(input)
```

```
    if (input<="Z"and input>="A")or(input<="z"and input>="a"):
```

```
        return True
```

```
    return False
```

'''

isNum 判断是否为数字

'''

```
def isNum(input):
```

```

input=str(input)
if input<="9"and input>="0":
    return True
return False
'''
stateChange 用于基于 输入状态 下一个字符的类型 返回下一个状态
'''
def stateChange(nowState:int,input:str,stateList:list,tempStr:str):
    # 多层处理
    print(' ')
    print('nowState    input    tempStr')
    print(nowState,'    ',input,'    ',tempStr)
    if nowState==1 and input=="":
        nextState=1
    elif nowState == 1 and input == ' ':
        nextState = 1
    elif nowState == 1 and input is None:
        nextState = 1
    elif nowState==1 and isChar(input):
        tempStr+=input
        nextState=2
    elif nowState==1 and isNum(input):
        tempStr += input
        nextState=4
    elif nowState==1 and input=="<":
        nextState=8
    elif nowState==1 and input==">":
        nextState=11

    #单一符号处理
    elif nowState==1 and input=="-":
        stateList.append('MINU')
        nextState=1
    elif nowState==1 and input=="*":
        stateList.append('MULT')
        nextState=1
    elif nowState==1 and input=="/":
        stateList.append('DIV')
        nextState=1
    elif nowState==1 and input=="=":
        stateList.append('ASSIGN')
        nextState=1
    elif nowState==1 and input==";":
        stateList.append('SEMICN')

```

```

        nextState=1
    elif nowState==1 and input==",":
        stateList.append('COMMA')
        nextState=1
    elif nowState==1 and input=="(":
        stateList.append('LPARENT')
        nextState=1
    elif nowState==1 and input==)":
        stateList.append('RPARENT')
        nextState=1
    elif nowState==1 and input=="[":
        stateList.append('LBRACK')
        nextState=1
    elif nowState==1 and input=="]":
        stateList.append('RBRACK')
        nextState=1
    ""
    elif nowState==1 and input=="\"":
        stateList.append('PHASE')
        nextState=1
    ""
    elif nowState==1 and input=="(":
        stateList.append('COLON')
        nextState=1

# 第二层的计算 非终结
elif nowState==2 and (isChar(input)or isNum(input)):
    tempStr += input
    nextState=2
elif nowState==4 and isNum(input):
    tempStr += input
    nextState=4
elif nowState == 1 and input == "\"":
    stateList.append('PHASE')
    nextState = 6

# 第二层计算终结
elif nowState==2 and (isChar(input)==False and isNum(input)==False):
    stateList.append('CHARCON')
    stateList.append(tempStr)
    tempStr=""
    nowState=1
    nextState,tempStr=stateChange(nowState,input,stateList,tempStr)
elif nowState==4 and isNum(input)==False:

```

```

        stateList.append('INTCON')
        stateList.append(int(tempStr))
        tempStr = ""
        nowState = 1
        nextState, tempStr = stateChange(nowState, input, stateList, tempStr)
    elif nowState==8 and input==" ":
        stateList.append('LTOE')
        nextState=1
    elif nowState==8 and input!=" ":
        stateList.append('LSS')
        tempStr = ""
        nowState = 1
        nextState, tempStr = stateChange(nowState, input, stateList, tempStr)
    elif nowState==11 and input==" ":
        stateList.append('GTOE')
        nextState=1
    elif nowState==11 and input!=" ":
        stateList.append('GRE')
        tempStr = ""
        nowState = 1
        nextState, tempStr = stateChange(nowState, input, stateList, tempStr)
    elif nowState==6 and input=="\\":
        stateList.append('STRCON')
        stateList.append(tempStr)
        stateList.append('PHASE')
        tempStr=""
        nextState=1
    elif nowState==6 and input!="\\":
        tempStr+=input
        nextState=6
    else:
        print('unable to match')
        print(input)
        nextState = 1
    # 其他情况的处理          设定为不处理
    """
    else :
        print('unable to match')
        print(input)
        stateList.append(input)
        nextState=1
    """
    return nextState,tempStr

```



```

"""
strMatch
"""
def strMatch(input:str):
    stateList=list()
    nowState=1
    tempStr=""
    for i in range(len(input)):
        inputC=input[i]
        nowState,tempStr=stateChange(nowState, str(inputC), stateList, tempStr)
    return stateList

"""
strClean 该函数处理以下问题
{'name': '标识符', 'class': 'IDENFR'},
{'name': '字符常量', 'class': 'CHARCON'},
{'name': '字符串', 'class': 'STRCON'},
1.将 STRCON 从 CHARCON 中分出来
2.将 IDENFR 从 CHARCON 中分出来
"""
def strClean(input:list):
    input.append("")
    input.append("")

    for i in range(len(input)-2):
        if input[i]=='CHARCON':
            if input[i+1]=='const':
                input[i] == 'IDENFR'
                input[i+1]='CONSTTK'
                i-=1
            if input[i+1]=='int':
                input[i] == 'IDENFR'
                input[i+1]='INTTK'
                i-=1
            if input[i+1]=='if':
                input[i] == 'IDENFR'
                input[i+1]='IFTK'
                i-=1
            if input[i+1]=='else':
                input[i] == 'IDENFR'
                input[i+1]='ELSETK'
                i-=1
            if input[i+1]=='do':
                input[i] == 'IDENFR'

```

```

        input[i+1]='DOTK'
        i-=1
    if input[i+1]=='while':
        input[i] == 'IDENFR'
        input[i+1]='WHILETK'
        i-=1
    if input[i+1]=='for':
        input[i] == 'IDENFR'
        input[i+1]='FORTK'
        i-=1
    if input[i+1]=='scanf':
        input[i] == 'IDENFR'
        input[i+1]='SCANFTK'
        i-=1

    del input[len(input) - 1]
    del input[len(input) - 1]
    return input
'''
readFile 读取文件 以 str list 的形式输出
'''
def readFile(address=r"C:\Users\leilu\Desktop\input.txt"):
    with open(address, encoding='utf-8') as file:
        data = file.readlines()
        print(data)
    file.close()
    for i in range(len(data)):
        print(data[i])
    return data
'''
writeFile 书写文件 以 str list 的形式输出
'''
def writeFile(data:list,address=r"C:\Users\leilu\Desktop\output.txt"):
    with open(address, "w") as file:
        for i in range(len(data)):
            file.write(data[i])
            file.write("\r\n")
            print(data[i])
    file.close()
'''
handle 将以上的函数整合为可用
'''
def handle():
    data=readFile()

```

```

output=list()
for i in range(len(data)):
    tempStr=data[i]
    tempOut=strMatch(tempStr)
    tempOut=strClean(tempOut)
    tempSave=""
    for j in range(len(tempOut)):
        tempSave+=('['+str(tempOut[j])+']')
    output.append(str(tempSave))
writeFile(output)
if __name__ == '__main__':
    matchList=[
        #抽象类型
        {'name':'标识符','class':'IDENFR'},
        {'name':'整型常量','class':'INTCON'},
        {'name':'字符常量','class':'CHARCON'},
        {'name':'字符串','class':'STRCON'},

        # 字符常量转标识符
        {'name':'const','class':'CONSTTK'},
        {'name':'int','class':'INTTK'},
        {'name':'if','class':'IFTK'},
        {'name':'else','class':'ELSETK'},
        {'name':'do','class':'DOTK'},
        {'name':'while','class':'WHILETK'},
        {'name':'for','class':'FORTK'},
        {'name':'scanf','class':'SCANFTK'},

        # 单符号
        {'name':'-','class':'MINU'},
        {'name':'*','class':'MULT'},
        {'name':'/','class':'DIV'},
        {'name':'<','class':'LSS'},
        {'name':'>','class':'GRE'},
        {'name':'=','class':'ASSIGN'},
        {'name':';','class':'SEMICN'},
        {'name':',','class':'COMMA'},
        {'name':'(','class':'LPARENT'},
        {'name':')','class':'RPARENT'},
        {'name':'[','class':'LBRACK'},
        #双符号
        {'name':'<=','class':'LEQ'},
    ]
    handle()

```

二、实验结果及分析（包括结果描述、实验现象分析、影响因素讨论、综合分析和结论等）

1.测试集

如图，为一个斐波那契数列的 c 语言版本代码

```
input.txt - 记事本

文件  编辑  查看

#include<stdio.h>
int Fibon(int n)
{
    if (n == 1|| n == 2)
    {
        return 1;
    }
    else
    {
        return Fibon(n-1)+ Fibon(n-2);
    }
}
int main()
{
    for (int i = 1; i <= 10; i++)
    {
        printf("%d ", Fibon(i));
    }
    return 0;
}
```

2.程序读取

可见读取无异常

```
C:\Users\leilu\AppData\Local\Programs\Python\Python310\python.exe C:/Users/leilu/Desktop/编译原理实验/实验五/tes
['#include<stdio.h>\n', 'int Fibon(int n)\n', '{\n', '    if (n == 1|| n == 2)\n', '        {\n', '\t    return 1;\n', '\n', '        }\n', '    }\n', '    else\n', '    {\n', '        return Fibon(n-1)+ Fibon(n-2);\n', '    }\n', '}\n', 'int main()\n', '{\n', '    for (int i = 1; i <= 10; i++)\n', '    {\n', '        printf("%d ", Fibon(i));\n', '    }\n', '    return 0;\n', '}\n']
```

3.程序计算状态 和 储存字符串

```
test x
nowState    input    tempStr
2    c      in
nowState    input    tempStr
2    l      inc
nowState    input    tempStr
2    u      incl
nowState    input    tempStr
2    d      inclu
nowState    input    tempStr
2    e      includ
nowState    input    tempStr
```

4.从 CHARCON 中匹配标识符 STRCON

```
test x
1
unable to match
CHARCON      printf
CHARCON      Fibon
CHARCON      i
```

5.程序输出结果

```
test x
[CHARCON][include][LSS][CHARCON][stdio][CHARCON][h][GRE]
[CHARCON][INTTK][CHARCON][Fibon][LPARENT][CHARCON][INTTK][CHARCON][n][RPARENT]
[CHARCON][IFTK][LPARENT][CHARCON][n][ASSIGN][ASSIGN][INTCON][1][CHARCON][n][ASSIGN][ASSIGN][INTCON][2][RPARENT]
[CHARCON][return][INTCON][1][SEMICN]
[CHARCON][ELSETK]
[CHARCON][return][CHARCON][Fibon][LPARENT][CHARCON][n][MINU][INTCON][1][RPARENT][CHARCON][Fibon][LPARENT][CHARCON][n][MINU][INTCON][1][RPARENT]
[CHARCON][INTTK][CHARCON][main][LPARENT][RPARENT]
[CHARCON][FORTK][LPARENT][CHARCON][INTTK][CHARCON][i][ASSIGN][INTCON][1][SEMICN][CHARCON][i][LTOE][INTCON][10]
[CHARCON][printf][LPARENT][PHASE][STRCON][%d ][PHASE][COMMA][CHARCON][Fibon][LPARENT][CHARCON][i][RPARENT][RPA
```

6.检测 TXT 写入情况

```
output.txt - 记事本

[CHARCON][include][LSS][CHARCON][stdio][CHARCON][h][GRE]
[CHARCON][INTTK][CHARCON][Fibon][LPARENT][CHARCON][INTTK][CHARCON][n][RPA

[CHARCON][IFTK][LPARENT][CHARCON][n][ASSIGN][ASSIGN][INTCON][1][CHARCON][n]

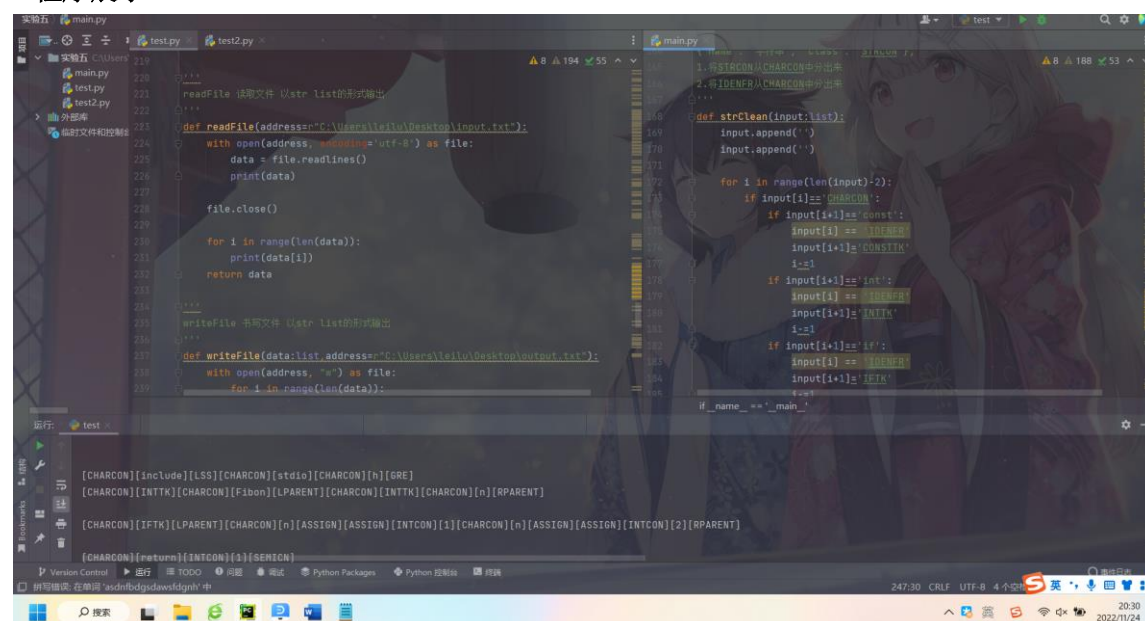
[CHARCON][return][INTCON][1][SEMICN]

[CHARCON][ELSETK]

[CHARCON][return][CHARCON][Fibon][LPARENT][CHARCON][n][MINU][INTCON][1][RPA

[CHARCON][INTTK][CHARCON][main][LPARENT][RPARENT]
```

7.程序展示



三、实验小结、建议及体会

【实验小结】

通过了这次的实验，本人学会了解了词法分析程序的设计与实现，并且通过了实践将自己的所学进行的实践。

同时也进一步的熟练了 Python 的使用。

【建议】

- 1.还行，就这还叫加分实验，这个我觉得比 13 简单好吧
- 2.这该死的表格能不能发一个电子版的，看着好难受啊。

【体会】

很简单的实验

