#include<iostream>

#include<string>

#include<fstream>

#include<regex>

#include<iomanip>

#include<queue>

#include<Windows.h>

#define max 10000

//正则表达式头文件

using namespace std;

typedef struct node;

typedef node \*tree;

int graph[max][max]; //函数嵌套定义邻接矩阵

int num[max];

int visited[max];

string name ("test.txt");

//括号匹配

int kuohao = 0;

//函数长度计数开关

int flag = 0;

//函数行数

int fun = 0;

//代码总行数

int line = 0;

struct result

{

int Code;//代码行

int Comments;//注释行

int Blanklines;//空行

int scode ;//代码行数

int scomment;//注释行数

int sspace;

int countfun ;//函数个数

int funlen ;//函数总长度

int maxlen;//最长函数

double avelen ;//函数平均长度

int maxline ;//最长函数所在行数

int maxh ;

string funname[max];

string maxfun;//最长函数名

}list;

string evaluate(result l)

//根据获取到的数据，对程序进行评分

{

string value;

if (l.avelen >= 10 && l.avelen <= 15)

value += 'A';

else if ((l.avelen >= 8 && l.avelen < 10) || (l.avelen > 15 && l.avelen <= 20))

value += 'B';

else if ((l.avelen >= 5 && l.avelen < 8) || (l.avelen > 20 && l.avelen <= 24))

value += 'C';

else

value += 'D';

if (list.scomment >= 15 && list.scomment <= 25)

value += 'A';

else if ((list.scomment >= 10 && list.scomment < 15) || (list.scomment > 25 && list.scomment <= 30))

value += 'B';

else if ((list.scomment >= 5 && list.scomment < 10) || (list.scomment > 30 && list.scomment <= 35))

value += 'C';

else

value += 'D';

if (list.sspace >= 15 && list.sspace <= 25)

value += 'A';

else if ((list.sspace >= 10 && list.sspace < 15) || (list.sspace > 25 && list.sspace <= 30))

value += 'B';

else if ((list.sspace >= 5 && list.sspace < 10) || (list.sspace > 30 && list.sspace <= 35))

value += 'C';

else

value += 'D';

return value;

}

/\*根据成绩给出相印的评语\*/

string remark(char s)

{

if (s=='A')

return " ";

else if (s == 'B')

return " ";

else if (s == 'C')

return " ";

else

return " ";

}

void analyze(string s) //判断是哪一类

{

//匹配到函数定义(regex\_match测试正则表达式是否与整个目标字符串相匹配)

if (regex\_match(s, regex("(\\w{1,10}) (\\w{1,100})\\(.{0,100}\\).{0,2}")))

//函数个数+1

{

list.countfun++;

//开启计数器

flag = 1;

}

if (regex\_search(s, regex("\\{")))

{

kuohao++;

}

if (regex\_search(s, regex("\\}")))

{

if (kuohao)

kuohao--;

if(kuohao==0&&flag)

{

fun++;

//加上尾花括号

list.funlen++;

fun = 0;

flag = 0;

}

}

if (regex\_search(s, regex("//")))

//注释行搜索;

list.Comments++;

if (regex\_search(s, regex("\\/\\\*")))

//注释行搜索;

list.Comments++;

else if (s == "")

//空行搜索

list.Blanklines++;

else {

if (flag)

{

list.funlen++;

}

list.Code++;

}

if (flag) {

fun++;

}

}

/\*展示面板\*/

void show()

{

list.scode = (int)((double)list.Code / (list.Code + list.Comments + list.Blanklines)\*100+0.5);

list.scomment = (int)((double)list.Comments / (list.Code + list.Comments + list.Blanklines) \* 100 + 0.5);

list.sspace = 100 - list.scode - list.scomment;

list.avelen = (double)list.funlen / list.countfun;

string res = evaluate(list);

cout << "The result of analysing program file \"" + name + "\":" << endl;

cout << " 代码行数 : " << list.Code << endl;

cout << " 注释行数 : " << list.Comments << endl;

cout << " 空行数 : " << list.Blanklines << endl;

cout << " 代码 注释 空行" << endl;

cout << " ==== ======== =====" << endl;

cout << " " <<list.scode<< "% "<<setw(6)<<list.scomment << "% " << list.sspace<<"%"<< endl;

cout << " 包含了 " << list.countfun << " 条函数." << endl;

cout << " 平均长度 " <<list.avelen<< " 行" << endl;

cout << " 代码等级 " << res[0] <<" "<< remark(res[0]) << endl;

cout << " 注释等级 " << res[1] <<" "<< remark(res[1]) << endl;

cout << " 空行等级 " << res[2] <<" "<< remark(res[2]) << endl;

}

int main()

{

int key;

cout << "---------------------------------------------------------------" << endl;

cout << " 请输入进行分析的程序文件路径及其后缀（可使用相对路径）" << endl;

cout << "---------------------------------------------------------------" << endl;

cout << "分析文件路径：";

getline(cin, name);

ifstream in(name);

//读入文件

system("cls");

if (!in)

{

cout << "----------------------------------------------------------" << endl;

cout << " 文件打开失败 " << endl;

cout << "----------------------------------------------------------" << endl;

Sleep(2000);

system("cls");

main();

}

string str;

while (1)

{

line++;

if (in.eof())

//判断文件读取结束eof有东西就不会返回false

break;

getline(in, str);

analyze(str);

}

in.close();

show();

cout << "----------------------------------------------------------" << endl;

cout << " 1.分析其他程序 2.退出 " << endl;

cout << "----------------------------------------------------------" << endl;

cout << "请输入序号 ： ";

cin >> key;

if (key == 1) {

system("cls"); //清屏

getchar(); //清除缓存

main();

}

return 0;

}