**西南大学计算机与信息科学学院**

**《C++》之**

**实验002： 射雕英雄传英雄出现次数**

**实验基本信息：**

未命名实验类型： □验证性 □设计性 □综合性

理论教师： 代立云 实验指导教师：代立云

学生姓名: 宋行健 学号: 222018321062006

班级： 18级软件工程一班

时间： 2019.5.17

**实验过程：**

1. 实验代码

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

using namespace std;

int main()

{

ifstream in; //读入文件流；

ofstream out; //写入文件流；

string name\_temp; //存循环时临时的英雄名称；

string line; //读入文件流，每次读一行，储存变量；

//把每个英雄罗列在一个vector里面；

vector<string> HeroName = { "郭靖","黄蓉","杨康","穆念慈","郭啸天","李萍","杨铁心","包惜弱","黄药师","柯镇恶","朱聪","韩宝驹","南希仁","张阿生","全金发","韩小莹" ,"欧阳锋","马钰","谭处端","刘处玄","丘处机","王处一","郝大通","孙不二","王重阳","周伯通","尹志平","程瑶迦","陈玄风","梅超风","陆乘风","冯默风","陆冠英","欧阳克","一灯大师","洪七公","梁长老","裘千仞","裘千丈","乔寨主","完颜洪烈","完颜洪熙","杨康","彭连虎","梁子翁","灵智上人","沙通天","侯通海","沈青刚","吴青烈" };

vector<int> Hero\_number(HeroName.size(), 0); //定义新向量，大小同英雄名字，初始值都为0；

cout << "射雕英雄传十大英雄：" << endl;

in.open("E:\\C++\\射雕英雄传.txt"); //打开要读入的文件；

if (!in)

cout << "Open failed(in)" << endl; //检验原文件是否打开成功；

else

{

while (getline(in, line)) //当in成行读入有数据，则将读入的数据存入line；

{

for (int i = 0; i < HeroName.size(); i++)

{

name\_temp = HeroName[i]; //临时存要找的英雄名字；

if (!line.find(name\_temp)) //如果line中含有英雄名称关键字计数变量+1；

{

Hero\_number[i] += 1;

}

}

}

}

in.close(); //关闭文件；

//冒泡排序排出前十位；

int number\_temp;

for (int i = 0; i < HeroName.size() - 1; i++)

{

for (int j = 0; j < HeroName.size() - 1 - i; j++)

{

if (Hero\_number[j] < Hero\_number[j + 1])

{

//对出现次数进行冒泡排序；

number\_temp = Hero\_number[j];

Hero\_number[j] = Hero\_number[j + 1];

Hero\_number[j + 1] = number\_temp;

//同时对其对应的英雄名字进行位置交换；

name\_temp = HeroName[j];

HeroName[j] = HeroName[j + 1];

HeroName[j + 1] = name\_temp;

}

}

}

//输出前十的英雄；

for (int i = 0; i < 10; i++)

{

cout << "TOP " << i + 1 << " :" << HeroName[i] << "\t" << Hero\_number[i] << "次" << endl;

}

ofstream svg;

float height[10];

svg.open("E:\\C++\\射雕英雄传svg.svg"); //创建一个svg文件

svg << "<?xml version=\"1.0\" encoding=\"GB2312\"?> " << endl; //设置为GB2312可以显示出中文

svg << "<svg width=\"800\" height=\"850\" xmlns=\"http://www.w3.org/2000/svg\">" << endl;

for (int i = 0; i < 10; i++)

{

height[i] = 0.5 \* Hero\_number[i]; //将出现次数转换为高度

}

for (int i = 0; i < 10; i++) //将高度表示为矩形的长度

{

svg << "<rect x=\"" << (i + 1) \* 70 << "\" y=\"" << 800 - (height[i]) << "\" width = \"50\" height = \"" << height[i] << "\" style =\"fill:blue;\"/> " << endl;

svg << "<text font-size=\"20\" x=\"" << (i + 1) \* 70 + 5 << "\" y=\"" << 780 - (height[i]) << "\">" << Hero\_number[i] << "</text>" << endl;

}

//画坐标轴

svg << "<line x1 = \"30\" y1 = \"800\" x2 = \"800\" y2 = \"800\" style=\"stroke:black; stroke - width:10\"/>" << endl; //X轴

svg << "<line x1 = \"30\" y1 = \"800\" x2 = \"30\" y2 = \"80\" style=\"stroke:black; stroke - width:10\"/>" << endl; //Y轴

//X轴箭头

svg << "<line x1 = \"800\" y1 = \"800\" x2 = \"780\" y2 = \"780\" style=\"stroke:black; stroke - width:10\"/>" << endl;

svg << "<line x1 = \"800\" y1 = \"800\" x2 = \"780\" y2 = \"820\" style=\"stroke:black; stroke - width:10\"/>" << endl;

//Y轴箭头

svg << "<line x1 = \"50\" y1 = \"100\" x2 = \"30\" y2 = \"80\" style=\"stroke:black; stroke - width:10\"/>" << endl;

svg << "<line x1 = \"10\" y1 = \"100\" x2 = \"30\" y2 = \"80\" style=\"stroke:black; stroke - width:10\"/>" << endl;

for (int i = 0; i < 10; i++) //标注横坐标人物

{

svg << "<text style=\"cursor: move; pointer-events: inherit;\" font-size=\"20px\" font-family=\"SimSun\" x=\"" << (i + 1) \* 70 << "\" y=\"820\">" << HeroName[i] << "</text>" << endl;

}

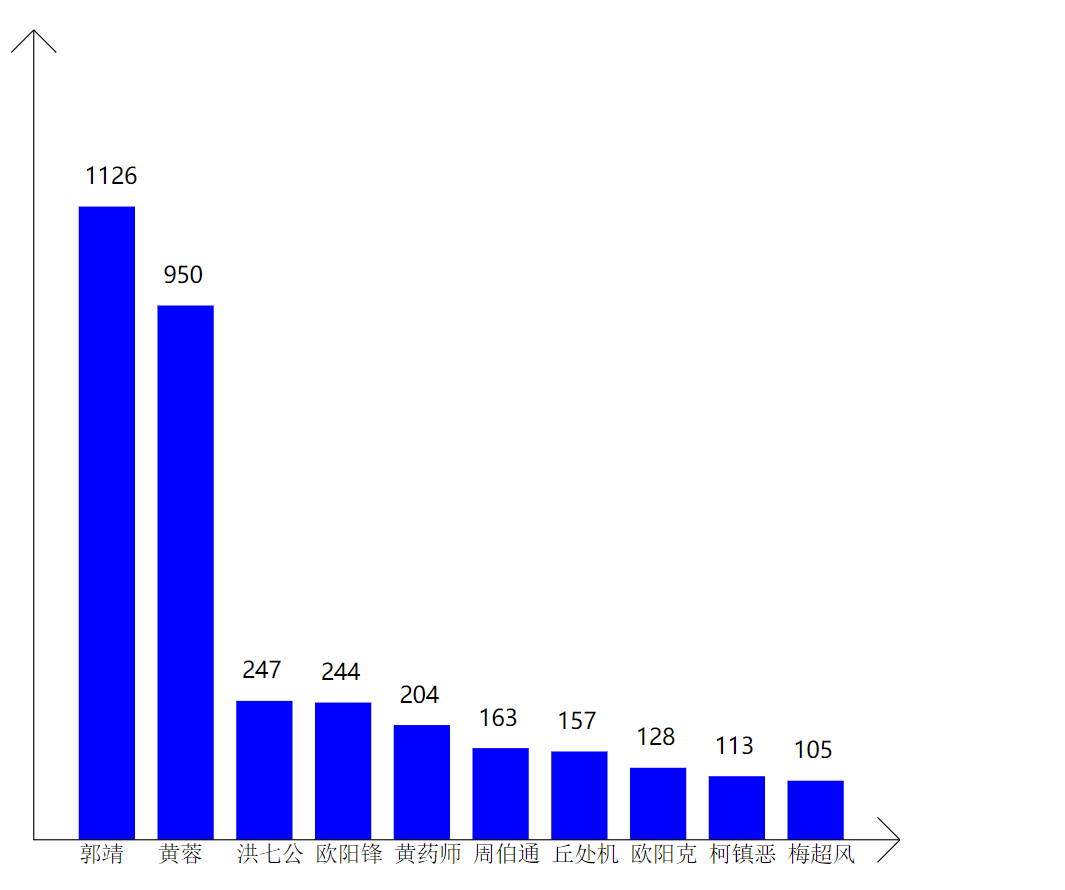
svg << "</svg>" << endl;

svg.close();

return 0;

}

1. SVG结果



1. 实验感想

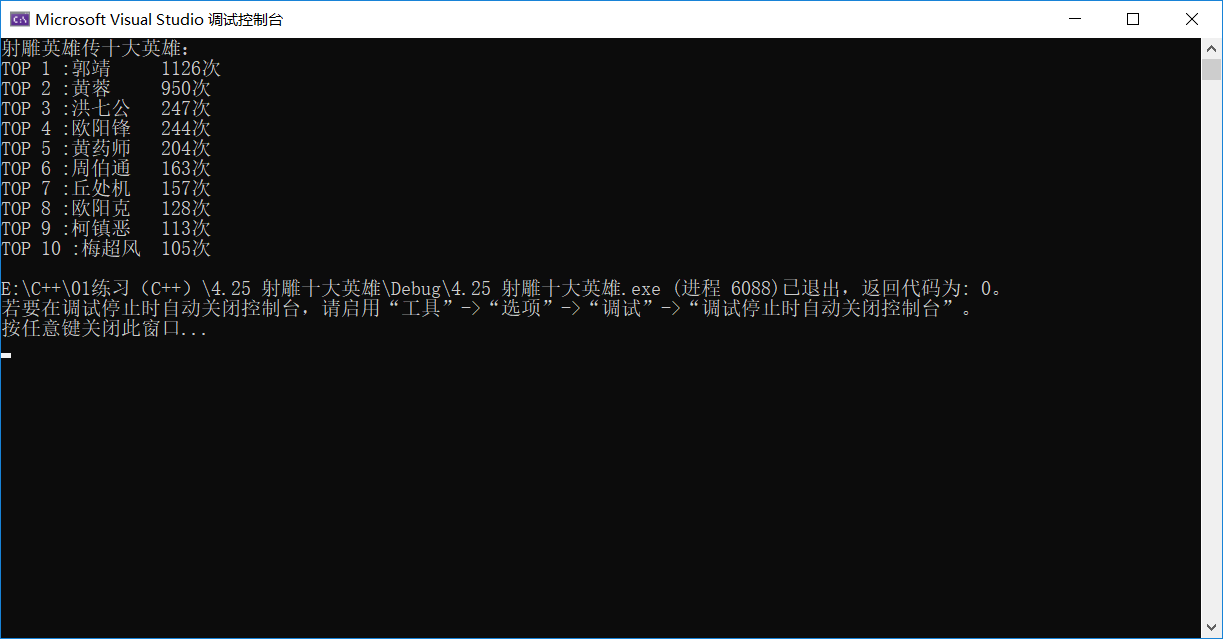
通过本次实验，我更加熟练掌握了文件读入读出流的应用。利用<fstream>包对“射雕英雄传.txt”文件进行逐行读取，在每一行的读取数据中寻找是否含有英雄名字的关键词，若含有，则将其对应的计数变量加一，当全文扫描完成后，对所存的计数变量进行“冒泡法排序”，排出前十位的英雄并将其输出。

在这个程序中，我主要用到了vector<string> HeroName和vector<int> Hero\_number (HeroName.size(),0)这两个向量组，一个用来储存英雄姓名字符串，另一个用来储存其对应的计数变量。

另外，在这个实验中再次使用find(name\_temp)函数对字符串进行查找。

在函数的最后，我创建了一个svg的文件，然后把数据转化为矩形的高度展现出来。

对于这样子初始化，vector<string> HeroName = { "郭靖","黄蓉","杨康" }; 在Dev-C++中编译不过，但是在Visual Studio 2019 中可以编译通过。

1. 实验截图