|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** | **王卓然** | **学号** | **20233005358** | **专业** | **计算机科学与技术** |

提示:交实验报告时，命名格式为：C++实验N-学号-姓名.docx， N为实验的序号。

# 实验8 文件操作

一、实验目的

1．理解文件的概念，熟悉C++文件及分类；熟悉C++的流类，掌握流对象的定义方法；

2．掌握C++文件的打开、文件的读/写及文件的关闭方法，熟悉其具体应用。

3．综合应用《高级语言程序设计》/《面向对象程序设计知识》知识，用文件方式保存数据，提高编程能力及程序调试能力。

二、实验学时

课内实验：2课时 课外练习：4课时

三 本实验涉及的新知识

㈠ 文件的概念

1．文件的访问格式

[盘符：][路径] 文件基本名.扩展名

其中，路径是用“\”分隔的一组目录，在C++中，由于“\”表示转义字符，因此在表示文件时，应在路径中使用“\\”。

2．C++中的文件及分类

⑴ C++中的文件由字符流或二进制流组成，称流式文件。

⑵ 按文件的组织形式，C++的文件分为字符文件(ASCII文件)及二进制文件。

⑶ 文件的基本操作包括文件的打开、文件的读/写、文件的关闭

㈡ 文件的打开与关闭

1．C++文件处理流类

⑴ C++中，文件处理用三个流类：

ifstream—文件输入流类，用于从文件中输入（读文件）；

ofstream—文件输出流类，用于输出到文件（写文件）；

fstream—文件输入/输出流类，用于从文件输入或输出到文件（读/写文件）。

⑵ 文件流对象的定义

若要打开文件进行读写操作，必须定义相应的流对象。

① 文件输入流对象的定义：ifstream 对象名；

② 文件输出流对象的定义：ofstream 对象名；

③ 文件输入/输出流对象的定义：fstream 对象名；

⑶ 说明：当在程序中进行文件操作时，应加上头文件“fstream.h”。

2．文件打开

文件打开通常用成员函数。

⑴ 用open( )函数打开

① 原型

void open( const char \*s,ios\_base::openmode ，mode=ios\_base::out|ios\_base::trunc)

其中，第一个参数表示打开的文件，第二个参数表示文件打开方式，第三个参数表示访问方式。

② 调用方法

**文件流对象.open(“文件名”[，ios\_base::打开方式][，访问方式])**

⑵ 在定义文件流对象时打开文件。

格式：

**文件流类名 文件流对象( “文件名”)**

用文件流对象打开文件后，该对象就代表了被打开的文件。

⑶ 文件打开方式

① app：以输出添加方式打开文件

② ate：文件打开后定位在文件尾

③ binary：以二进制方式打开，缺省时以文本方式

④ in：以只读方式打开文件

⑤ out：以只写方式打开文件

⑥ trunc：若打开已存在的文件进行写操作，则将清除文件原有内容变成空文件；若文件不存在则创建文件。

⑷ 说明

① 当用ifstream定义流对象并打开一个文件时，默认为ios\_base::in方式。

② 当用ofstream定义流对象并打开一个文件时，默认为ios\_base::out方式。

③ 当用fstream定义流对象并打开一个文件时，应给出打开方式。

④ 可用“位或”运算（“|”），实现以多种方式打开文件。

3．文件异常处理

打开一个文件可能失败，在程序中通常应加上异常处理程序。

⑴ 用条件语句判断文件流标志位“failbit”是否为true。

格式1：

**if (!文件流对象)**

**{<异常处理程序>}**

格式2：

**if(文件流对象)**

**{<正常处理程序>}**

⑵ 用成员fail()函数

格式1：

**if(文件流对象.fail())**

**{<异常处理程序>}**

格式2：

**if(!文件流对象.fail())**

**{<正常处理程序>}**

4．文件的关闭

**格式：文件流对象.close();**

㈢ 文件的读写

1．文本文件的读写

⑴ 读文件格式

**文件输入流对象.get(char);**

⑵ 写文件格式

**文件输出流对象.put(char ch);**

2．二进制文件的读写

二进制文件的读/写分别用成员函数read( )、write( )实现。

⑴ 读二进制文件的格式

**输入文件流对象.read((char\*)&对象或&对象数组名[下标]，sizeof(对象名或所属类名))；**

⑵ 写二制文件的格式：

**输出文件流对象.write((char\*)&对象或&对象数组名[下标]，sizeof(对象名或所属类名))；**

四、实验内容

㈠ 验证及认知实验

按要求调试下列程序并回答相关问题。

程序1（exp\_801.cpp）

#include<iostream>

#include<fstream>

using namespace std;

void main(void)

{ ifstream f1;

f1.open("d:\\vcprg\\e301.cpp");

if(!f1)

{ cout <<"文件打开失败（文件不存在），按任意建结束！";

cin.get();

return;

}

char c;int n=0;

while(f1)

{ f1.get(c);

cout<<c;

if(c=='\n') n++;

}

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

f1.close();

}

问题：

1. 调试运行程序将显示 文件打开失败（文件不存在），按任意建结束！ 。
2. 在D盘下建立vcprg文件夹，将e301.cpp文件拷贝在该文件夹，再重新运行程序。该程序的功能是 从e301.cpp文件读入字符，并记录行数 。
3. 程序中的“ifstream f1;”的作用是 建立文件输入流对象f1 ，“f1.open("d:\\vcprg\\e301.cpp");” 的作用是 打开e301.cpp ，“f1.get(c);”的作用是 读入字符 ；“f1.close();”的作用是 关闭文件 。
4. 程序中的变量n统计的是 行数（文件中含有’\n’的数量） 。

程序2（exp\_802.cpp）

#include<iostream>

#include<fstream>

using namespace std;

void main(void)

{ ofstream f;//定义文件输出流对象

f.open("d:\\vcprg\\test.txt");

// f.open("d:\\vcprg\\test.txt",ios\_base::app);

if(!f)

{ cout <<"建立文件失败（文件夹“vcprg”不存在），按任意建结束！";

cin.get();

return;

}

char c;

cout<<"输入文件的内容('#'结束):"<<endl;

while((c=cin.get())!='#')

{ f<<c;}

f.close();

c=cin.get();

cout<<"按任意键显示文件内容："<<endl;

c=cin.get();

ifstream f1; //定义文件输入流对象

f1.open("d:\\vcprg\\test.txt"); //打开文件

while(f1) //判断文件未结束则循环

{ f1.get(c);

cout<<c;

}

cout<<endl;

f1.close();//关闭文件

}

问题：

1. 运行两次程序，观察程序的输入过程及显示结果，再将“f.open("d:\\vcprg\\test.txt");”改为“f.open("d:\\vcprg\\test.txt",ios\_base::app);”，再重新运行程序，比较其差异。

ios\_base::app是添加到原文件末尾，原来的会覆盖文件内容

1. 该程序的功能是 从用户那里获取输入，将输入写入到一个文件中，然后读取该文件的内容并显示给用户。 。
2. 程序中“f<<c;”的作用是 将用户的输入字符写入到文件中。 。

㈡ 知识应用实验

完善、调试通过下列程序，并按所要求回答问题。

程序3（exp\_803.cpp）

//程序功能：将d盘vcprg文件夹下的文件e301.cpp复制为text.txt

#include<iostream>

#include<fstream>

using namespace std;

void main(void)

{ (1) ; //定义文件输入流对象f1

(2) ; //定义文件输出流对象f2

f1.open("d:\\vcprg\\e301.cpp"); //以读方式打开文件

if( (3) )

{ cout <<"文件打开失败（文件夹不存在），按任意建结束！";

cin.get();

return;

}

f2.open("d:\\vcprg\\text.txt"); //以写方式打开文件

char c;

while( f1 ) //判断文件未结束则循环

{ ( 4 ) ; //从文件输入流对象f1中读取一个字符到字符变量c

( 5 ) ; //将字符变量c的值写入到文件输出流对象f2中

}

cout<<"文件复制成功"<<endl;

f1.close();//关闭文件

f2.close();

}

问题：程序中

（1）处应为 ifsteam f1 ，（2）处应为 ofstream f2

（3）处应为 !f1 ，（4）处应为 f1 >> c

（5）处应为 f2 << c

程序4（exp\_804.cpp）

//程序功能：输入多个person记录于“personal.dat”中，然后再显示输出

#include<iostream.h>

#include<fstream.h>

#include"person.h"

void main(void)

{ char na[11],se[3];

int y,m,d;

person psn;

ifstream f1;

ofstream f2;

f2.open("d:\\vcprg\\personal.dat",ios::binary|ios::app);

if(!f2)

{ cout<<"\a can't open personal.dat";

return;

}

char ch='y';

while(ch=='y')

{ cout<<"请输入学生信息:\n";

cout<<"姓名:";cin>>na;

cout<<"性别:";cin>>se;

cout<<"出生年:";cin>>y;

cout<<"出生月:";cin>>m;

cout<<"出生日:";cin>>d;

psn.set\_person(na,se,y,m,d);

f2.write((6));//将psn对象写入文件中

cout<<"继续吗(y/n)? ";cin>>ch;

}

f2.close();

f1.open("d:\\vcprg\\personal.dat",ios::binary);

if(!f1) //或者用if(f1.fail())

{ cout<<"\a can't open personal.dat";

return;

}

while(!f1.eof())

{ f1.read((7));//将文件中的一条记录读到psn中

cout<<psn.get\_name()<<"\t";

cout<<psn.get\_sex()<<"\t";

cout<<psn.get\_year()<<"\t";

cout<<psn.get\_month()<<"\t";

cout<<psn.get\_day()<<"\t";

cout<<endl;

}

f1.close();

}

问题：程序中

（7）处应为 f2.write((char\*)&psn, sizeof(psn));

，（8）处应为 f1.read((char\*)&psn. sizeof(psn));

㈢ 程序设计实验

1．缩写程序，对文件进行以下操作：

⑴ 建立文本文件；

⑵ 添加文件内容；

⑶ 显示文件内容；

⑷ 显示文件自然段数；

⑸ 显示文件行数；

⑹ 检查文件中某一符号是否匹配。

要求：

① 文本文件名自定；

② 用菜单方式实现；

③ 检查文件中某一符号是否匹配时，用“实验十”中的堆栈类（“tmpstack.h”），匹配符号包括“（”与“）”、“{”与“}”、“[”与“]”等，匹配符号在键盘上输入。

④ 程序存放在exp\_1205.cpp中。

#include <iostream>  
#include <fstream>  
#include <stack>  
#include <string>  
#include <algorithm>  
using namespace std;  
void FileCreate(const string& filename){  
 ofstream file(filename);  
 file.close();  
}  
void FileWrite(const string& filename, const string& text){  
 ofstream file(filename, ios::app);  
 file << text;  
 file.close();  
}  
void displayContent(const string& filename) {  
 ifstream file(filename);  
 string line;  
 while (getline(file, line)) {  
 cout << line << endl;  
 }  
 file.close();  
}  
int countParagraphs(const string& filename) {  
 ifstream file(filename);  
 string line;  
 int count = 0;  
 bool newParagraph = true;  
 while(getline(file, line)){  
 if (line.empty()) {  
 newParagraph = true;  
 } else if (newParagraph) {  
 count++;  
 newParagraph = false;  
 }  
 }  
 file.close();  
 return count;  
}  
int countLines(const string& filename) {  
 ifstream file(filename);  
 string line;  
 int count = 0;  
 while(getline(file, line)){  
 count++;  
 }  
 file.close();  
 return count;  
}  
bool checkSymbols(const string& filename) { *//没见着哪有堆栈类，直接使用std::stack* ifstream file(filename);  
 char c;  
 stack<char> stack;  
 while(file.get(c)){  
 if (c == '(' || c == '{' || c == '[') {  
 stack.push(c);  
 } else if (c == ')' || c == '}' || c == ']') {  
 if (stack.empty()) {  
 return false;  
 }  
 char open = stack.top();  
 stack.pop();  
 if ((c == ')' && open != '(') || (c == '}' && open != '{') || (c == ']' && open != '[')) {  
 return false;  
 }  
 }  
 }  
 file.close();  
 return stack.empty();  
}  
int main(){  
 string fileName = "text.txt";  
 FileCreate(fileName);  
 FileWrite(fileName, "Hello, World!\n\n");  
 displayContent(fileName);  
 cout << "Paragraphs: " << countParagraphs(fileName) << endl;  
 cout << "Lines: " << countLines(fileName) - 1 << endl;  
 cout << "Symbols: " << (checkSymbols(fileName) ? "correct" : "incorrect") << endl;  
 return 0;  
}

2．编写程序，用文件实现对学生数据的处理。包括：

⑴ 学生数据输入

⑵ 学生数据显示

⑶ 学生数据查询

要求：

① 用菜单方式实现；

② 学生类用前面实验中的“student.h”定义的学生类，重载“<<”、“>>”实现学生数据的输入/输出（输出时各列对整齐）；

③ 学生数据查询按学号或姓名查询；

④ 程序存放在exp\_1206.cpp中。

#include <iostream>  
#include <fstream>  
#include <vector>  
#include <algorithm>  
using namespace std;  
  
class Student{  
public:  
 string name;  
 int id;  
 int score;  
 Student(string n = "", int sc = 0, int i = 0) : name(n), score(sc), id(i) {}  
 friend istream& operator>>(istream& in, Student& s){  
 in >> s.name >> s.score >> s.id;  
 return in;  
 }  
 friend ostream& operator<<(ostream& out, Student& s){  
 out << s.name << " " << s.score << " " << s.id;  
 return out;  
 }  
};  
  
void saveToFile(vector<Student>& students) {  
 ofstream file("students.txt");  
 for (auto& student : students) {  
 file << student << endl;  
 }  
 file.close();  
}  
  
void loadFromFile(vector<Student>& students) {  
 ifstream file("students.txt");  
 Student student;  
 while (file >> student) {  
 students.push\_back(student);  
 }  
 file.close();  
}  
  
int main(){  
 vector<Student> students;  
 loadFromFile(students);  
 int choice;  
 do{  
 cout << "--------------------------------" << endl;  
 cout << "1. 添加学生" << endl;  
 cout << "2. 展示学生信息" << endl;  
 cout << "3. 通过ID查找学生信息" << endl;  
 cout << "4. 通过姓名查找学生信息" << endl;  
 cout << "5. 退出" << endl;  
 cout << "--------------------------------" << endl;  
 cout << "请选择: ";  
 cin >> choice;  
  
 switch (choice) {  
 case 1: {  
 system("cls");  
 Student student;  
 cout << "输入学生信息(姓名 成绩 ID)：";  
 cin >> student;  
 students.push\_back(student);  
 saveToFile(students);  
 break;  
 }  
 case 2:  
 system("cls");  
 for (auto& student : students) {  
 cout << student << endl;  
 }  
 break;  
 case 3: {  
 system("cls");  
 int id;  
 cout << "输入学生ID: ";  
 cin >> id;  
 vector<Student>::iterator it = find\_if(students.begin(), students.end(), [&](Student& student) {  
 return student.id == id;  
 });  
  
 if (it != students.end()) {  
 cout << \*it << endl;  
 } else {  
 cout << "未找到ID为'" << id << "'的学生"<< endl;  
 }  
 break;  
 }  
 case 4: {  
 system("cls");  
 string name;  
 cout << "输入学生姓名: ";  
 cin >> name;  
 vector<Student>::iterator it = find\_if(students.begin(), students.end(), [&](Student& student) {  
 return student.name == name;  
 });  
  
 if (it != students.end()) {  
 cout << \*it << endl;  
 } else {  
 cout << "未找到姓名为'" << name << "'的学生"<< endl;  
 }  
 break;  
 }  
 case 5:  
 cout << "Exiting...\n";  
 break;  
 default:  
 cout << "Invalid choice. Please try again.\n";  
 break;  
 }  
 }while(choice != 5);  
  
 return 0;  
}

五、实验收获与创新

1．自已拟定一个解决实际问题的题目，综合应用《高级语言程序设计》、《面向对象程序设计》中的知识，编程实现。

要求：至少定义两个类，数据用文件保存。

2．通过实验一到实验十二，谈谈你对面向对象程序设计的认识。