**广州大学学生实验报告**

**开课学院及实验室：**计算机学院/电子信息楼418A **2017年11月29日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学 院** | 计算机 | **年级/专业/班** | 软件171 | **姓名** | 谢金宏 | **学号** | 1706300001 |
| **实验课程名称** | 程序设计基础 | | | | | **成绩** |  |
| **实验项目名称** | **实验5文件（多文件程序结构）** | | | | | **指导老师** | 张为 |

(\*\*\*（1）报告只能为文字和图片；（2）实验项目名称不能有任何改动；（3）必须填写正确的姓名和学号；否则会影响最后成绩的统计，后果自负\*\*\*)

**一、实验目的及要求**

目的：掌握多文件程序结构的基本操作流程，熟悉有关操作的常用函数的使用，并能够根据实际编程需要进行灵活运用。

要求： 能够运用多文件程序完成一个完整的程序功能操作流程。

**二、实验设备与平台**

1. 实验设备：计算机；

2. 平台：Windows 10操作系统，GCC 7.2.0。

**三、实验内容及步骤**

**实验5\_1：**输入一串字符，再输入一个指定字符，将字符串中删去指定字符再显示出来。要求程序把输入字符串、删除指定字符和显示字符串的功能分别在各自单独的文件由外部函数来处理，主文件中的主函数只作调用各外部函数实现程序功能。

**Main.cpp**

#include "player.h"

#include "recorder.h"

#include "adapter.h"

const int LEN = 100;

int main()

{

char buf[LEN];

// 读取用户的字符串

play("[PLEASE GIVE ME SOME INPUT]", true);

play(">> "); record(buf, LEN);

play("[YOUR INPUT]", true);

play("<< "); play(buf, true);

// 读取用户指定的字符

char ch;

play("[PLEASE GIVE ME ONE CHARACTER] ");

play(">> "); record(ch);

play("[YOUR INPUT] ");

play(ch, true);

// 在字符串中删去字符

play("[PROCEEDING...]", true);

adapt(buf, ch);

// 输出修改后的字符串

play("<< "); play(buf, true);

play("[Goodbye~] : )", true);

}

**Player.h**

// 显示字符串。当endl为真时，附加换行符

void play(const char \* str, bool endl = false);

// 显示字符和对应的机器编码。当endl为真时，附加换行符

void play(char ch, bool endl = false);

**Player.cpp**

#include <cstdio>

#include <iostream>

void play(const char \* str, bool endl = false)

{

std::cout << str;

if (endl) std::cout << std::endl;

}

void play(char ch, bool endl = false)

{

printf("%c[%d]", ch, (int)ch);

if (endl) std::cout << std::endl;

}

**Recorder.h**

// 将用户输入保存在指定的字符中，忽略空白字符

void record(char &ch);

// 将用户输入保存在指定的字符串中，最多读取(length - 1)个字符

void record(char \* str, int length);

**Recorder.cpp**

#include <cstdio>

#include <cstring>

#include <iostream>

void record(char &ch)

{

std::cin >> ch;

}

void record(char \* str, int length)

{

memset(str, '\0', length);

for (int i = 1; i < length; i++)

{

int ch; ch = getchar();

if (ch != EOF && ch != '\n') {

\*str++ = ch;

}

else {

str++;

break;

}

}

\*str = '\0';

}

**Adapter.h**

// 在str中删除由ch指定的字符，返回删除字符的个数

int adapt(char \* str, char ch);

**Adapter.cpp**

#include <cstring>

#include <algorithm>

// 删除str中指定位置的字符，并将字符串中删除字符后的字符前移

void trim(char \* pos)

{

\*pos = '\0';

while (\*++pos != '\0') {

std::swap(\*pos, \*(pos - 1));

}

}

int adapt(char \* str, char ch)

{

int cnt = 0; char \* pos;

while ((pos = strchr(str, ch)) != NULL) {

trim(pos);

cnt++;

}

return cnt;

}