

程序设计训练 日记与报告

专业： 通信工程
班级： 181 班
学号： 201811050962
姓名： 陈呈涛
指导教师： 刘荣香 易宏杰

信息与控制工程学院

2020 年 12 月

目录

实习日记.....	1
程序设计训练报告.....	6
一、 基础部分.....	6
1. 函数问题（数的合并）.....	6
2. 线性表问题（猴子选大王）.....	9
3. 排序问题（奖学金）.....	13
4. 字符串问题.....	16
5. 栈问题（多项式计算器）.....	21
6. 面向对象问题.....	25
二、 综合部分.....	29
轿车信息管理系统.....	29
1. 系统需求分析.....	29
2. 总体设计.....	31
3. 详细设计.....	33
4. 系统实现与调试.....	38
5. 系统使用测试说明.....	48
6. 总结.....	51
参考文献.....	53

实习日记

实习日记	2020 年 12 月 21 日
<p>今天是 2020 年 12 月 21 日，是程序设计训练的第一天。</p> <p>这次的程序设计训练是利用我们以前学过的 C 语言，数据结构，数据库，以及面向对象程序设计等课程进行简单的程序设计，比之前的更加综合。</p> <p>首先老师给我们开了一个动员大会，老师给我们讲解了程序设计的一些注意事项，程序设计训练的任务和报告的书写方式以及分组答辩的一些事项，之后我们就到了信控大楼 515 进行程序设计训练了，我们便开始了为期两周的程序设计训练。</p> <p>刚开始对程序训练的任务很恐惧，感觉哪一个也不会，后来仔细读题没有想象的那么难，对一基础题的第一题要定义 fun 函数通过三种不同的方式来实现，由于对这三种方式已经遗忘，变得不知所措，经过在网上查阅资料又明白了很多知识，这使得对做程序设计训练的任务更加顺手。经过两个小时的操作，稍微有点思路了。</p>	
实习日记	2020 年 12 月 22 日
<p>今天是 2020 年 12 月 22 日，是程序设计训练的第二天。</p> <p>由于前些日子我们没有课，每天早上都要睡个懒觉，现在突然早起，背着有点沉重的电脑去有点远的信控大楼进行程序设计训练，想到这属实是不想起。</p> <p>到了信控大楼开始了今天的程序设计训练，今天接着昨天的继续做，经过我的一顿操作基础题的第一题解决了，瞬间得到了满足感，提高了对这次程序设计训练自己的兴趣。</p> <p>由于兴趣的提好和第一题训练使得基础题第二题变得容易了一些，虽说是容易些但是也存在很多问题，这个题要求是用数组和链表来实现，当我看到这个题的时候我的思路就只有数组的方法，链表的方法经过和同学讨论我有了思路，到下课使劲我也做完了，第二题就这样磕磕绊绊做完了。</p>	

实习日记

2020 年 12 月 23 日

今天是 2020 年 12 月 23 日是程序设计训练的第三天。

今天做的是基础部分的第三题和第四题，第三题是排序问题，看到这个题看时感觉很简单，你有多想就直接做了，虽说能与运行，但是不能达到任务书上的要求，任务书上的要求是，如果两个同学总分相同，再按语文成绩从高到低排序，如果两个同学总分和语文成绩都相同，那么规定学号小的同学排在前面，我的程序只符合前两项，后一项不能实现，经过的仔细的思考 and 修改实现了。

第四题是字符串问题，由于在学字符串的时候没有好好学，现在遇到了有点头疼，只好硬上了，做着做着感觉也没有这么难，不过也遇到了一点小问题，通过伟大的网络不一会就解决了，今天完成了两道题很高兴。

实习日记

2020 年 12 月 24 日

今天是 2020 年 12 月 24 日，是程序设计训练的第四天。

今天和前几天不一样，由于要对 DSP 进行答辩没有很快进入状态，因为报告还没有做完，在想报告怎么做，在老师看不到还偷偷在做 DSP 的报告，这样做的不只是我一个人，被老师看到了，老师说做 DSP 也没问题，只要不是来玩的就行，只要有事情做就行。老师说了这话之后我便放心的去做了，不一会儿就做完了。

就这样我就开始了今天的正式的程序设计训练，今天做的是基础题的第五题栈问题，设计多项式计算器，遇到运算符就停止运行，多次尝试和修改都不行，定义栈的存在问题，经过多次的修改能运行的但是计算结果是错误的。通过网上查资料学习栈的定义并且请教同学帮忙，换一个思路来做题，修改代码，多次测试最终成功。今天也算是完美的一天。

实习日记

2020 年 12 月 25 日

今天是 2020 年 12 月 25 日，程序设计训练的第五天。

今天做的是基础部分的最后一题面向对象问题，要求汽车类 `vehicle`，它具有一个需传递参数的构造函数，类中的数据成员：车轮个数 `wheels` 和车重 `weight` 放在保护段中；小车类 `car` 是它的私有派生类，其中包含载人数 `passenger_load`；卡车类 `truck` 是 `vehicle` 的私有派生类，其中包含载人类 `passenger_load` 和载重量 `payload`，并且每个类都有相关数据的输出方法。看到这个题目没有一点思路，瞬间没有了动力玩了一会手机，有一个同学的一句话提醒了我，我开始了今天的程序设计训练。编写完程序之后发现有错误建的类出现错误，不能实现小汽车 `car`、卡车 `truck` 是汽车类的公有派生类构造的函数不能传递参数，发现问题后寻找错误，由于是惯性思维的原因没有找的原因，通过和同学讨论找到问题，建的类存在问题，通过修改得以实现功能。

实习日记

2020 年 12 月 28 日

今天是 2020 年 12 月 28 日，是程序设计训练的第二周周一。

经过周末使得刚刚要习惯早起的我又陷入睡懒觉之中。今天就要开始了我们的综合部分的题目，我们组的题目是第九题，轿车信息管理系统，由计算机来代替人工执行一系列诸如显示、查找、添加、修改及删除等操作。要求轿车信息包括轿车的编号、型号、颜色、生产厂家、出厂日期、价格等；输入功能可以一次完成若干条信息的输入；显示功能能够完成全部轿车信息的显示（一屏最多显示 10 条，超过十条应能够自动分屏显示）；查找功能：可以按编号、名称等多种方式查找轿车信息还可以增加、删除、修改轿车信息等。我们今天的任务主要是完成我们组的分工，以及搜集和讨论有关轿车信息的资料。经过搜集和讨论我们有了整体的思路。

实习日记

2020 年 12 月 29 日

今天 2020 年 12 月 29 号，是程序设计训练的第二周周二。

今天我们的任务是对系统进行系统需求分析，汽车管理系统可由由计算机来代替人工对汽车信息进行输入、显示、查找、删除、修改等操做；一直以来人们使用传统人工的方式管理汽车的各种信息，这种管理方式存在着许多缺点,如效率低、另外时间一长,将产生大量的文件和数据,这对于查找、更新和维护都带来了不少的困难。随着科学技术的不断提高,计算机科学日渐成熟,其强大的功能已为人们深刻认识,它已进入人类社会的各个领域并发挥着越来越重要的作用。汽车信息管理系统是由计算机来代替人工执行一系列诸如输入、显示、查找、删除、修改等操作，以达到更加快速准确的实现你想实现的功能。并且在组内讨论画出了系统流程图。

实习日记

2020 年 12 月 30 日

今天是 2020 年 12 月 30 日，是程序设计训练的第二周周三。

今天我们完成的任务是轿车信息管理系统的总体设计部分和详细设计部分，要对系统功能进行分析，输入功能要求轿车信息包括轿车的编号、型号、颜色、生产厂家、出厂日期、价格等并且输入功能可以一次完成若干条信息的输入；显示功能能够完成全部轿车信息的显示(一屏最多显示 10 条,超过十条应能够自动分屏显示)；查找功能：可以按编号、名称等多种方式查找轿车信息，还可以增加、删除、修改轿车信息等。我们系统进行各个模块的划分与设计，分为输入模块，显示模块，查询模块，删除模块，和修改模式并画出了各个模块的系统功能模块图。画完这些我们就做 E—R 图和数据库表了，数据库表包括了编号，型号，颜色生产厂家，生产日期以及价格。今天做的这些中间虽然磕磕绊绊的事很充实。

实习日记

2020 年 12 月 31 日

今天是 2020 年 12 月 31 日，程序设计训练第二周的周四，也是最后一天。

由于今天下午我们有微博与天线的考试考试说今天我们不进行答辩了，答辩的时间定在了下周，具体时间再通知。我今天就开始了轿车信息管理系统的最重要的一部分，编写代码，让 Mysql 和 VS 建立连接，在 VS 中新建的 Mysql 结构体 mysql 和 sock, 通过 Mysql 所带的函数接口 mysql_init 和 mysql_real_connect 来实现 Mysql 和 VS 的建立连接, 从而实现系统类中 init() 函数的构建。链接完之后我又定义了 sys 类, 在 say 类中的用户可访问程序接口有 init() 初始化函数 function() 功能选择函数, secede() 系统函数退出, Input() 输入信息函数, display() 显示信息函数, search() 查找信息函数, deletel() 删除信息函数 modify() 修改信息函。再通过上述的函数通过 SQL 语句和 mysql_query 函数接口运行 SQL 语句实现各个功能。今天虽然没能够把代码写完，但是也收获也很多，剩余的代码利用元旦的时间来完成吧！

程序设计训练报告

一、基础部分

1. 函数问题（数的合并）

问题描述

编写一个能将两个数合并成一个数的函数 `fun`，具体合并方式是：将两个两位数的正整数 `a` 和 `b` 合并形成一个整数放在 `c` 中，并将 `a` 的十位和个位依次存放在 `c` 数的十位和千位上，`b` 的十位和个位依次存放在 `c` 数的个位和百位上，基本要求：在主函数中调用 `fun` 函数，`fun` 函数要写出以下三种：（1）通过返回值将新数 `c` 传给主函数；（2）通过引用参数将新数传给主函数；（3）通过指针型参数将新数传给主函数。

输入格式：

一行，包含两个两位整数 `a` 和 `b`。

输出格式：

一行，合并后的新数 `c`。

分析比较三种方式的区别。

问题分析：

定义一个 `fun` 函数，把每个数的个位，十位提取出来，按照要求乘以不同的倍数再求和，最后在主函数中调用 `fun` 函数

代码：

（1）通过返回值将新数 `c` 传给主函数

```
#include <iostream>
#include <ostream>
using namespace std;
int main()
{
    int fun(int a, int b);
    int a, b, c;
```



```

    cout << "输入a,b:" << endl;
    cin >> a >> b;
    c = fun(a, b);
    cout << "c=" << c << endl;
    return 0;
}
int fun(int a, int b)
{
    return (a % 10 * 1000 + b % 10 * 100 + a / 10 * 10 + b / 10);
}

```

(2) 通过引用参数将新数传给主函数

```

#include <iostream>
#include <ostream>
using namespace std;
int main()
{
    int fun(int&a,int&b,int&c);
    int a,b,c;
    cout<<"输入a,b:"<<endl;
    cin>>a>>b;
    fun(a,b,c);
    cout<<"c="<<c<<endl;
    return 0;
}
int fun(int&a,int&b,int&c)
{
    c=a%10*1000+b%10*100+a/10*10+b/10;
    return 0;
}

```

(3) 通过指针型参数将新数传给主函数

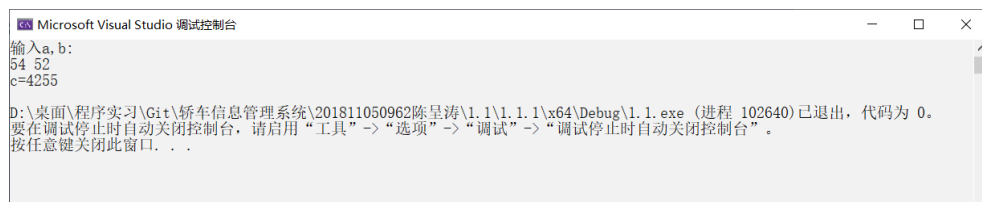
```

#include <iostream>
#include <ostream>
using namespace std;
int main()
{
    int fun(int*a,int*b,int*c);
    int a,b,c;
    cout<<"输入a,b:"<<endl;
    cin>>a>>b;
    fun(&a,&b,&c);
    cout<<"c ="<<c<<endl;
    return 0;
}
int fun(int*a,int*b,int*c)
{
    *c=*a%10*1000+*b%10*100+*a/10*10+*b/10;
    return 0;
}

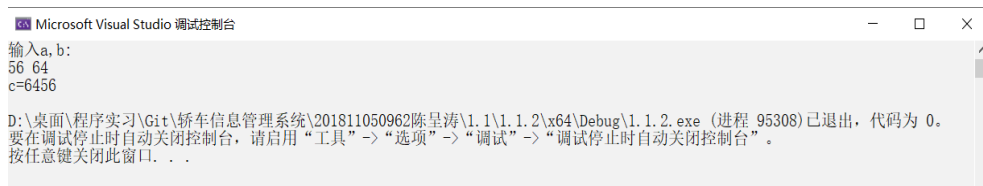
```

运行截图：

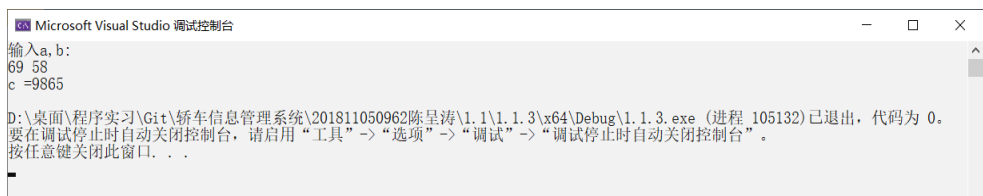
(1) 通过返回值将新数 c 传给主函数



(2) 通过引用参数将新数传给主函数



(3) 通过指针型参数将新数传给主函数



总结

编写代码时有几个地方输入错误，有几处漏掉；和{}等符号。运行代码，通过 VS 中解决方案中错误代码出现的行数，按照行数找到代码错误进行修改。

2. 线性表问题（猴子选大王）

问题描述：

一堆猴子都有编号，编号是 1，2，3 ...m，这群猴子（m 个）按照 1-m 的顺序围坐一圈，从第 1 开始数，每数到第 N 个，该猴子就要离开此圈，这样依次下来，直到圈中只剩下最后一只猴子，则该猴子为大王。

基本要求：

- （1） 输入数据：输入 m,n m,n 为整数， $n < m$
- （2）中文提示按照 m 个猴子，数 n 个数的方法，输出为大王的猴子是几号，建立一个函数来实现此功能
- （3）分别用数组和链表来实现，并比较分析不同存储结构下的时间复杂度和空间复杂度。

问题分析

（1）用数组实现

在一个数组中，数组中用 1 表示猴子在圈中，用 0 表示猴子已经出圈，数组下标对应与猴子编号对应，若一只猴子出圈，则将对应的数组值置为 0；在报数过程中，要略过值为 0 的猴子。最后循环找出那个地址的数据即为王，输出数据即可。

（2）用链表实现

最后一只指向第一只，建立猴子围成的圆圈，从 p1 开始，先找到第 n-1 只，下一只将是被淘汰的，被淘汰的从圆圈中删除，再次最后一只指向第一只围成新的圆圈，被删除的下一个为新一轮的新起点，循环 m-1 次，淘汰 m-1 只猴子，最后循环找出那个地址的数据即为王，输出数据即可。

代码：

（1）用数组实现

```
#include <stdio.h>

#define MaxSize 100

void king(int m, int n)
{
    int p[MaxSize];
```

```

int i,j,t=0;
for (i=0;i< m;i++) //构建初始序列，记录m只猴子在p[0]~p[m-1]中
    p[i]=i+1; //首次报数的起始位置为0
printf("大王的编号是:");
for (i=m;i>=1;i--)//循环要执行m次，有m个猴子要出圈；共有i从m开始递减
至1，i还表示在圈中猴子的数目
{
    t=(t+n-1)%i; //从t开始数1，其后第n-1个将数到n，t加n-1用%i取余，目
的是到达最后一个猴子可以折回去继续数
    for (j=t+1;j<=i-1;j++)//后面的元素前移一个位置，删除了编号为p[t]的猴
子
        p[j-1]=p[j];
}
printf("%d",p[t]);
}
int main()
{
    int m,n;
    printf("输入m,n\n");
    scanf_s("%d%d",&m,&n);
    king(m,n);
    return 0;
}

```

（2）用链表实现

```

#include <iostream>
using namespace std;
struct Monkey
{
    int num;//猴子的编号

```

```

    struct Monkey* next; //下一只猴子
};

int main()
{
    int m, n, i, j, king;
    Monkey *head,*p1,*p2;
    cout <<"输入m,n"<< endl;
    cin>>m>>n;
    if (n==1)
    {
        king=m;
    }
    else
    {
        //建立猴子围成的圆圈
        p1 = p2 = new Monkey;
        head = p1;
        p1->num = 1;
        for (i = 1; i < m; i++) //其余m-1只猴子
        {
            p1 = new Monkey; //p1是新增加的
            p1->num = i + 1;
            p2->next = p1;
            p2 = p1;          //p2总是上一只
        }
        p2->next = head;      //最后一只再指向第一只，成了一个圆圈

        //开始数了
        p1 = head;
    }
}

```

```

for (i = 1; i < m; i++) //循环m-1次，淘汰m-1只猴子
{
    //从p1开始，数n-1只就找到第n只了
    for (j = 1; j < n - 1; j++) //实际先找到第n-1只，下一只将是被淘汰
    的

        p1 = p1->next;    //围成圈的，可能再开始从第一只数，如果
    还未被淘汰的话

        //找到了
        p2 = p1->next; //p2将被删除

        //cout<<"第"<<i<<"轮淘汰"<<p2->num<<endl; //可以这样观察中
    间结果

        p1->next = p2->next; //p2就这样被“架空了”
        p1 = p2->next; //下一轮数数的新起点
        delete p2; //将不在链表中的结点放弃掉
    }

    king = p1->num;

    delete p1;
}

cout << king << endl;

return 0;
}

```

运行截图：

(1) 用数组实现

```

Microsoft Visual Studio 调试控制台
输入m, n
89 98
大王的编号是:26
D:\桌面\程序实习\Git\轿车信息管理系统\201811050962陈呈涛\1.2\1.2数组\x64\Debug\1.2数组.exe (进程 112004) 已退出，代码为 0
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .

```

(2) 用链表实现

```

Microsoft Visual Studio 调试控制台
输入m, n
96 65
19
D:\桌面\程序实习\Git\轿车信息管理系统\201811050962陈呈涛\1.2\1.2链表\x64\Debug\1.2.exe (进程 107792) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .

```

总结

在做用数组实现的时候，我的算法有问题无论输入什么样的数据输出结果都是同一个数，请求同学的帮忙，给同学说了我的想法，同学帮我找出了里面的错误，在同学的帮助下最终能够实现了。编写代码时有几个地方输入错误，有几处漏掉；和{}等符号。运行代码，通过 VS 中解决方案中错误代码出现的行数，按照行数找到代码错误的地方进行修改。

3. 排序问题（奖学金）

问题描述

某小学最近得到了一笔赞助，打算拿出其中一部分为学习成绩优秀的前 55 名学生发奖学金。期末，每个学生都有 33 门课的成绩:语文、数学、英语。先按总分从高到低排序，如果两个同学总分相同，再按语文成绩从高到低排序，如果两个同学总分和语文成绩都相同，那么规定学号小的同学 排在前面，这样，每个学生的排序是唯一确定的。

任务：先根据输入的 33 门课的成绩计算总分，然后按上述规则排序，最后按排名顺序输出前五名名学生的学号和总分。注意，在前 55 名同学中，每个人的奖学金都不相同，因此，你必须严格按上述规则排序。例如，在某个正确答案中，如果前两行的输出数据(每行输出两个数:学号、总分) 是：

7 279

5 279

这两行数据的含义是:总分最高的两个同学的学号依次是 7 号、5 号。这两名同学的总分都是 279(总分等于输入的语文、数学、英语三科成绩之和)，但学号为 7 的学生语文成绩更高一些。如果你的前两名的输出数据是：

5 279

7 279

则按输出错误处理，不能得分。

输入：

第 1 行为一个正整数 n ，表示该校参加评选的学生人数。

第 2 到 $n+1$ 行，每行有 3 个用空格隔开的数字，每个数字都在 0 到 100 之间。第 j 行的 3 个数字依次表示学号为 $j-1$ 的学生的语文、数学、英语的成绩。每个

学生的学号按照输入顺序编号为 1~n (恰好是输入数据的行号减 1)。

所给的数据都是正确的，不必检验。

输入样例：

8

80 89 89

88 98 78

90 67 80

87 66 91

78 89 91

88 99 77

67 89 64

78 89 98

输出：

共有 5 行，每行是两个用空格隔开的正整数，依次表示前 5 名学生的学号和总分。

输出样例：

8 265

2 264

6 264

1 258

5 258

基本要求：使用结构体实现

问题分析

先定义一个结构体，将学号、语文成绩、数学成绩、英语成绩、总成绩放入其中;再定义一个按照要求排序的函数，先比较总分，若总分相同，再比较语文成绩，分数高的在前，分数低的在后；若总成绩和语文成绩都相同则序学号小的再前，学号大的在后。

代码：

```
#include <iostream>
```

```
#include <algorithm>
```



```

using namespace std;

struct student
{
    int chinese; //语文成绩
    int math; //数学成绩
    int english; //英语成绩
    int num; //学号
    int sum = 0; //总分
}s[301];

int cmp(student p1, student p2)
{//结构体
    if (p1.sum == p2.sum && p1.chinese != p2.chinese) { //总分相同，语文成绩高排在前
        return p1.chinese > p2.chinese;
    }
    if (p1.sum == p2.sum && p1.chinese == p2.chinese) { //总分相同，语文成绩相同，序号小在前
        return p1.num < p2.num;
    }
    else { //否则总分越高在越前
        return p1.sum > p2.sum;
    }
}

int main()
{
    int n; //人数
    cout << "输入人数及每个人的语文、数学、英语的成绩" << endl;
    cin >> n;
    for (int i=1; i<=n; i++)
    {

```

```

s[i].num=i;

cin>>s[i].chinese>>s[i].math>>s[i].english;

s[i].sum=s[i].chinese+s[i].math+s[i].english;
}

sort(1+s,1+n+s,cmp);//排序

cout<<"前五名的学号和成绩分别是"<<endl;

for (int i = 1; i <= 5; i++)

{//输出前5名

    cout<<s[i].num<<" "<<s[i].sum<<endl;

}

return 0;

}

```

运行截图：

```

选择Microsoft Visual Studio 调试控制台
输入人数及每个人的语文、数学、英语的成绩
5
98 99 100
98 88 99
98 100 99
99 98 100
98 90 97
前五名的学号和成绩分别是
4 297
1 297
3 297
2 285
5 285
D:\桌面\程序实习\Git\轿车信息管理系统\201811050962陈呈涛\1.3\x64\Debug\1.3.exe (进程 85032) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口。 . . .

```

总结

在定义按要求排序的函数时，思路有点混乱，多次尝试不能实现想要的结果，排序的算法存在问题，最终通过在网上查阅有关的算法，以及和同学讨论，经过多次修改代码得以实现题目要求的功能

4. 字符串问题

问题描述

疫情期间，瘟疫公司为了配合国家进行疫情防控工作，开始了对传染人数预测的研究。据悉，我国完备的医疗健康监测体系已经准确地绘制了 P 省感染分布情况地图。地图是一个 $N \times M$ 的矩形，地图上的每个点有三种状态未感染、被感

染和被保护，分别用大写字母 O,X,P 表示。例如：

OOOOOPOOOO

OXOOOPOOOO

OOOOOPOOOO

OOOOOPOOOO

之所以能根据监测数据成功预测，是因为某医疗科学团队经过实验推理证明，一段病毒周期之内一个感染区域会感染上、下、左、右四个方向上的区域，被保护的区域则不会被传染。例如，经历了 1 段病毒周期后的 P 省感染分布图变为：

OXOOOPOOOO

XXXOOPOOOO

OXOOOPOOOO

OOOOOPOOOO

但是随着时间的推移，2 段、3 段、4 段.....很多段病毒周期过后，病毒的最大感染情况为：

XXXXXPoooo

XXXXXPoooo

XXXXXPoooo

XXXXXPoooo

因为被保护区不会被感染，能将病毒的传染阻断。现在，作为瘟疫公司的高级工程师，你要实现的是输入一个地图和经历的病毒周期数，推算出经历了这些病毒周期后的新地图。

输入

共 $1+M+1$ 行

第一行：两个数字 N 和 M，表示地图的长宽。

第二行开始的 M 行：每行是长度为 M 的字符串，仅可能包含三个字母 O、X 或 P，表示地图上每个区域的状态。

第 M+2 行：一个整数 C，表示经历了 C 个病毒周期。

输出

共 M 行

表示经历了 C 个病毒周期过后新的感染情况分布地图

输入样例：

10 4

O O O O O P O O O O

O X O O O P O O O O

O O O O O P O O O O

O O O O O P O O O O

2

输出样例：

X X X O O P O O O O

X X X X O P O O O O

X X X O O P O O O O

O X O O O P O O O O

问题分析

定义一个字符串数组，即题目中的地图，在数组中输入字符串并保存在数组中，对数组中的每一个字符串进行判断是不是等于 X，若等于 X 则对其上下左右进行判断是不是不等于 P 或者 X，若是不等于 P 或者 X 则令其为 A 再次执行对数组中的每一个字符串进行判断是不是等于 A，若等于 A 则令其等于 X。经历的病毒周期数是几就执行上述操作几次。

代码：

```
#include<iostream>
#include<ostream>
using namespace std;
int main()
{
    char a[100][100];
    int n,m,i,j,c,t;
    cout <<"输入n*m的矩形"<< endl;
    cin>>n>>m;
```

```

for (int i=1;i<=m;i++)
{
    for (int j=1;j<=n;j++)
    {
        cin>>a[i][j];
    }
}

cout << "输入经历的周期" << endl;

cin>>c;

for (int t = 1; t <= c; t++)
{
    for (int i = 1; i <= m; i++)
    {
        for (int j = 1; j <= n; j++)
        {
            if (a[i][j] == 'X')
            {
                if (a[i + 1][j] != 'P' && a[i + 1][j] != 'X')
                    a[i + 1][j] = 'A';
                if (a[i - 1][j] != 'P' && a[i + 1][j] != 'X')
                    a[i - 1][j] = 'A';
                if (a[i][j + 1] != 'A' && a[i + 1][j] != 'X')
                    a[i][j + 1] = 'A';
                if (a[i][j - 1] != 'P' && a[i + 1][j] != 'X')
                    a[i][j - 1] = 'A';
            }
        }
    }

    for (int i = 1; i <= m; i++)
    {

```


其为 A 时，不能实现题目所说的功能，要再次执行对字符串数组中的每一个字符串进行判断是不是等于 A,若其值等于 A 则令其等于 X。

5. 栈问题（多项式计算器）

问题描述

啊~多么美好的一个后序遍历问题...(疯狂暗示)本来要出另外一个题的...结果生成数据的时候翻车了就明天再出吧。今天先水一道多项式计算器。可不要在网上抄代码哦，一眼就看出来了。

输入一个简单多项式，输出结果。多项式可能包含对整数的加减两种运算，中间没有空格。

输入

一行，一个多项式，保证输入合法

输出

一个结果

输入样例：

$((1+2)*3+4)+5+6*7$

输出样例：

60

问题分析

（1）建立两个栈，一个用来存储操作数，另一个用来存储运算符，开始时在运算符栈中先压入‘/0’，一个表达式的结束符。

（2）然后从左至右依次读取表达式中的各个符号（操作数或者运算符）；

（3）如果读到的是操作数直接存入操作数栈；

（4）如果读到的是运算符，则作进一步判断：

①若读到的是‘/0’结束符，而且此时运算符栈的栈顶元素也是‘/0’结束符，则运算结束，输出操作数栈中的元素即为最后结果。

②若读到的是‘（’或者读到的运算符的优先级比目前的运算符栈中的栈顶元素的优先级高，则将运算符直接存入运算符栈，继续读表达式中的下一个符号，重复步骤（3）和（4）；

③若读到的是‘）’，而且此时运算符栈的栈顶元素是‘（’结束符，则将运算符栈

中的栈顶元素退出来，继续读表达式中的下一个符号，重复步骤（3）和（4）；

④若读到的运算符的优先级等于或小于之前的运算符的优先级，则从操作数中退出 2 个，从运算符中退出一个进行运算，将运算结果存入操作数栈；再把之前读到的运算符与目前的运算符栈顶比较，重复步骤（4）

⑤若读到的运算符的优先级等于或小于之前的运算符的优先级，则从操作数中退出 2 个，从运算符中退出一个进行运算，将运算结果存入操作数栈；再把之前读到的运算符与目前的运算符栈顶比较，重复步骤（4）

代码：

```
#include <iostream>
#include <stack>
using namespace std;
double toNum(char* s, int& k)
{
    int flag=0;
    double x=0.0,y=0.1;
    while (s[k]>='0'&&s[k]<='9' || s[k]=='.')
    {
        if (s[k]>='0'&&s[k]<='9')
        {
            if (flag==0)
                x=x*10+s[k]-'0';
            else
            {
                x=x+y*(s[k]-'0');
                y=y*0.1;
            }
        }
        else
        {
            flag=1;
```



```

        }
        k=k+1;
    }
    return x;
}

int priority(char c)
{
    int k;
    switch (c)
    {
        case '*':k = 2; break;
        case '/':k = 2; break;
        case '+':k = 1; break;
        case '-':k = 1; break;
        case '(':k = 0; break;
        case ')':k = 0; break;
        default:k = -1; break;
    }
    return k;
}

int main()
{
    stack<double>sv;
    stack<char>sp;
    char c;
    int k=0,flag=1;
    double x,y;
    sp.push('\0');
    char*s=new char[100];

```

```

cout<<"输入多项式: "<<endl;
cin>>s;
c=s[k];
while (flag)
{
    if (c>='0'&& c<='9' || c=='.')
    {
        sv.push(toNum(s,k));
    }
    else if (c=='\0'&&sp.top()=='\0')
    {
        flag = 0;
    }
    else if (c=='(' || (priority(c)>priority(sp.top()))))
    {
        sp.push(c);
        k++;
    }
    else if (c == ')' && sp.top() == '(')
    {
        sp.pop();
        k++;
    }
    else if (priority(c) <= priority(sp.top()))
    {
        x=sv.top();
        sv.pop();
        y=sv.top();
        sv.pop();
        c=sp.top();
    }
}

```

```

        sp.pop();
        switch (c)
        {
            case '+':y = x + y; break;
            case '-':y = y - x; break;
            case '*':y = x * y; break;
            case '/':y = y / x; break;
        }
        sv.push(y);
    }
    c=s[k];
}

cout<<"=";

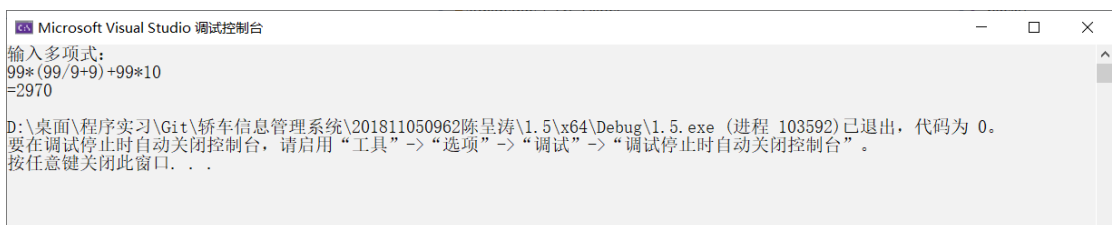
cout<<sv.top()<<endl;

return 0;

}

```

运行截图：



总结

遇到运算符就停止运行，多次尝试和修改都不行，定义栈的存在问题，经过多次的修改能运行的但是计算结果是错误的。通过网上查资料学习栈的定义并且请教同学帮忙，换一个思路来做题，修改代码，多次测试最终成功。编写代码时有几个地方输入错误，有几处漏掉；和{}等符号，运行代码，通过 VS 中解决方案中错误代码出现的行数，按照行数找到代码错误的地方进行修改。

6. 面向对象问题

问题描述

编写一个程序，有一个汽车类 `vehicle`，它具有一个需传递参数的构造函数，类中的数据成员：车轮个数 `wheels` 和车重 `weight` 放在保护段中；小车类 `car` 是它的私有派生类，其中包含载人数 `passenger_load`；卡车类 `truck` 是 `vehicle` 的私有派生类，其中包含载人类 `passenger_load` 和载重量 `payload`。每个类都有相关数据的输出方法。

问题分析

定义一个汽车类 `vehicle`，它的数据成员包括车轮个数和车；小车类 `car` 是它的公有派生类，新增数据成员载人数；卡车 `truck` 是汽车类的公有派生类，新增数据成员载人数和载重量。先把父类的属性进行了私有化再提供给子类调用，小车类 `car`、卡车类 `truck` 继承 `Vehicle` 父类。

代码：

```
#include<iostream>

#include <math.h>

using namespace std;

class vehicle//汽车类
{
protected:
    int wheels;// 车轮数
    int weight;// 重量
public:
    vehicle(int a, int b)
    {
        wheels = a;
        weight = b;
    }
    void show()
    {
        cout << "车轮数: " << wheels << "个" << endl;
        cout << "车 重: " << weight << "Kg" << endl;
```

```

    }
};

class car :private vehicle//小汽车类
{
    int passenger_load;//载人数
public:
    car(int a, int b, int passenger1) :vehicle(a, b)
    {
        passenger_load = passenger1;
    }
    void show()
    {
        vehicle::show();
        cout << "载人数: " << passenger_load << "人" << endl;
    }
};

class truck :private vehicle//卡车类
{
    int passenger_load;//载人数
    int payload;//载重量
public:
    truck(int a, int b, int passenger1, int payload1) :vehicle(a, b)
    {
        passenger_load = passenger1;
        payload = payload1;
    }
    void show()
    {
        vehicle::show();

```

```

        cout << "载人数: " << passenger_load << "人" << endl;
        cout << "载重量: " << payload << "Kg" << endl;

    }
};

int main()
{
    int wheels, weight, wheels_c, weight_c, passenger_load_c, wheels_t, weight_t,
    passenger_load_t, payload_t;

    cout << "输入汽车类的车轮个数、车重: " << endl;
    cin >> wheels >> weight;
    cout << "输入小车类的车轮个数、车重、载人数: " << endl;
    cin >> wheels_c >> weight_c >> passenger_load_c;
    cout << "输入卡车类的车轮个数、车重、载人数、载重量: " << endl;
    cin >> wheels_t >> weight_t >> passenger_load_t >> payload_t;
    cout << "\n" ;

    cout << "汽车类: " << endl;
    vehicle v(wheels, weight);
    v.show();
    cout << "\n";

    cout << "小车类: " << endl;
    car c(wheels_c, weight_c, passenger_load_c);
    c.show();
    cout << "\n";

    cout << "卡车类: " << endl;
    truck t(wheels_t, weight_t, passenger_load_t, payload_t);
    t.show();
}

```

```
cout << "\n";

}
```

运行截图：

```
Microsoft Visual Studio 调试控制台
输入汽车类的车轮个数、车重：
4 400
输入小车类的车轮个数、车重、载人数：
6 800 8
输入卡车类的车轮个数、车重、载人数、载重量：
6 700 6 1700

汽车类：
车轮数：4个
车 重：400Kg

小车类：
车轮数：6个
车 重：800Kg
载人数：8人

卡车类：
车轮数：6个
车 重：700Kg
载人数：6人
载重量：1700Kg

D:\桌面\程序实习\Git\轿车信息管理系统\201811050962陈呈涛\1.6\Debug\1.6.exe (进程 109012) 已退出，代码为 0。
要在调试停止时自动关闭控制台，请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...
```

总结

建的类出现错误，不能实现小汽车 car、卡车 truck 是汽车类的公有派生类构造的函数不能传递参数，发现问题后寻找错误，由于是惯性思维的原因没有找的原因，通过和同学讨论找到问题，建的类存在问题，通过修改得以实现功能。

二、综合部分

轿车信息管理系统

设计一个轿车信息管理系统，由计算机来代替人工执行一系列诸如显示、查找、添加、修改及删除等操作。

功能要求

- (1) 轿车信息包括轿车的编号、型号、颜色、生产厂家、出厂日期、价格等；
- (2) 输入功能：可以一次完成若干条信息的输入。
- (3) 显示功能：完成全部轿车信息的显示(一屏最多显示 10 条,超过十条应能够自动分屏显示)
- (4) 查找功能：可以按编号、名称等多种方式查找轿车信息
- (5) 增加、删除、修改功能：完成轿车信息的多种更新

1.系统需求分析

- (1) 系统功能需求

汽车管理系统可由计算机来代替人工对汽车信息进行输入、显示、查找、删除、修改等操做。

系统流程图

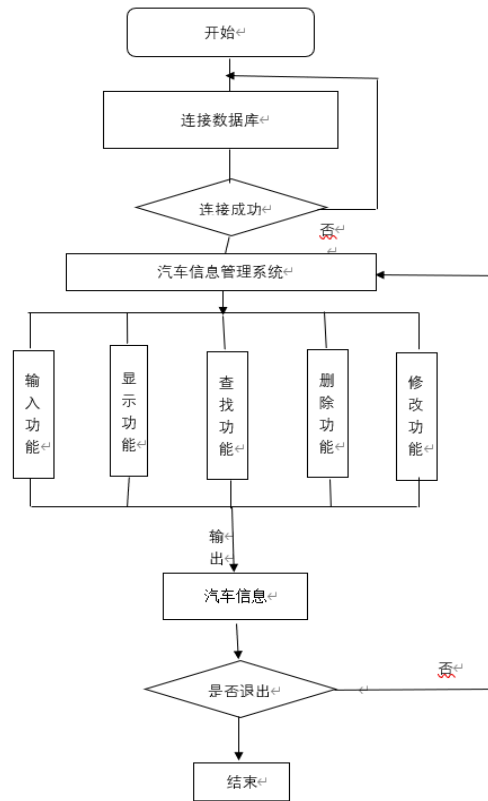


图 1 系统流程图

(2) 输入输出的要求

输入功能：可以一次完成若干条信息的输入。

输出功能：完成全部轿车信息的显示，一屏最多显示 10 条,超过十条应能够自动分屏显示。

(3) 系统设计目的与意义

一直以来人们使用传统人工的方式管理汽车的各种信息，这种管理方式存在着许多缺点,如效率低、另外时间一长,将产生大量的文件和数据,这对于查找、更新和维护都带来了不少的困难。

随着科学技术的不断提高,计算机科学日渐成熟,其强大的功能已为人们深刻

认识,它已进入人类社会的各个领域并发挥着越来越重要的作用。汽车信息管理系统是由计算机来代替人工执行一系列诸如输入、显示、查找、删除、修改等操作,以达到更加快速准确的实现你想实现的功能。

2. 总体设计

(1) 系统功能分析

汽车管理系统主要实现以下功能:

- ①输入功能: 可以一次完成若干条信息的输入。
- ②显示功能: 完成全部轿车信息的显示
- ③查找功能: 可以按编号、名称等多种方式查找轿车信息
- ④增加、删除、修改功能: 完成轿车信息的多种更新

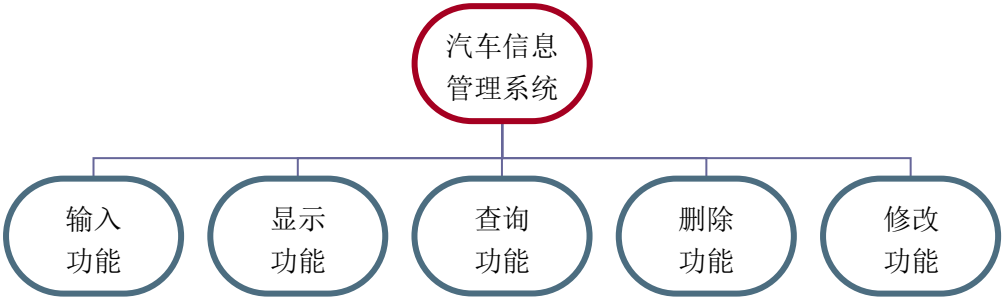


图 2 系统的整体结构

(2) 系统功能模块划分与设计 (系统功能模块图)

①输入功能模块

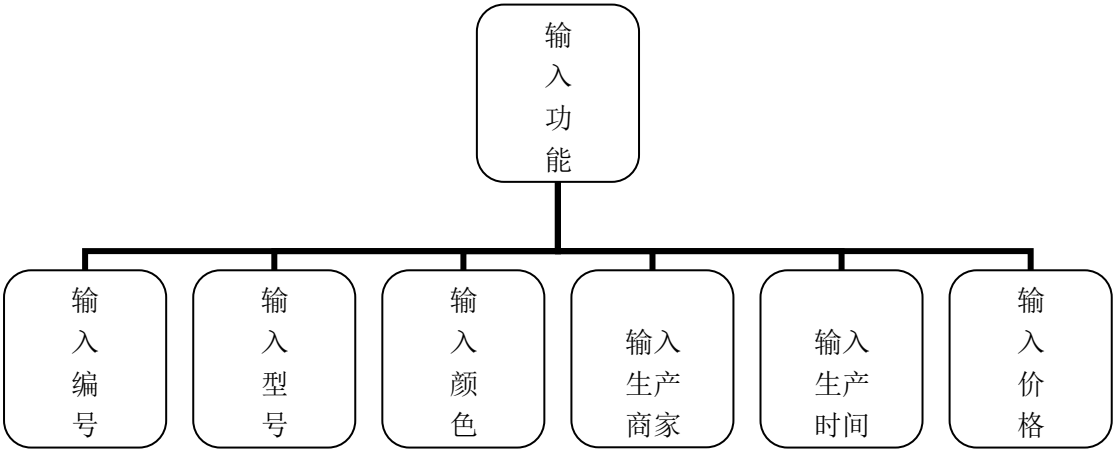


图 3 输入功能模块图

②显示功能模块

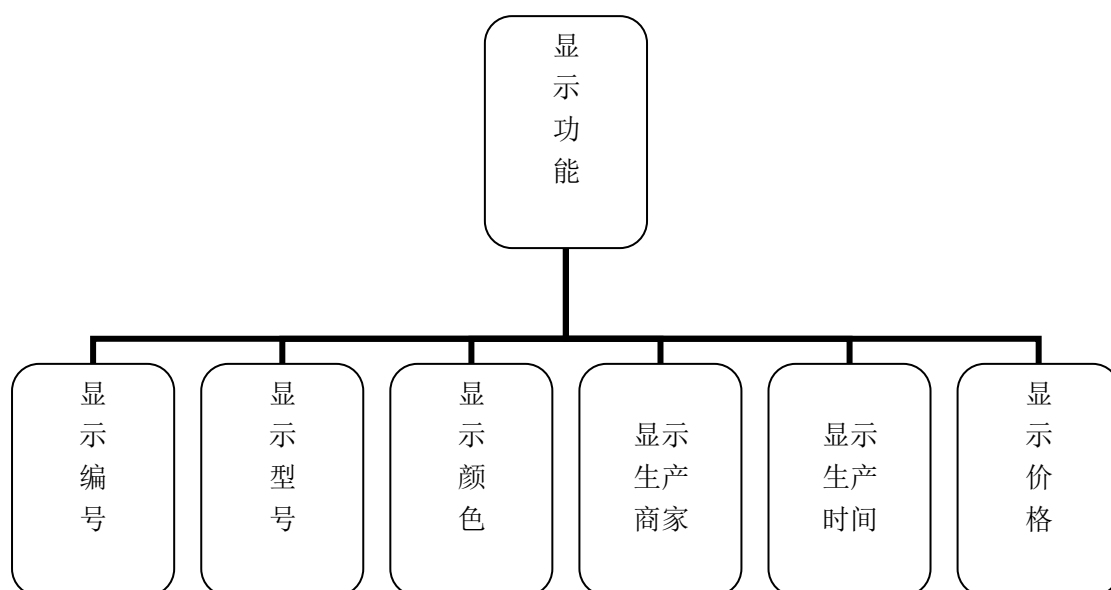


图 4 显示功能模块图

③查找功能模块

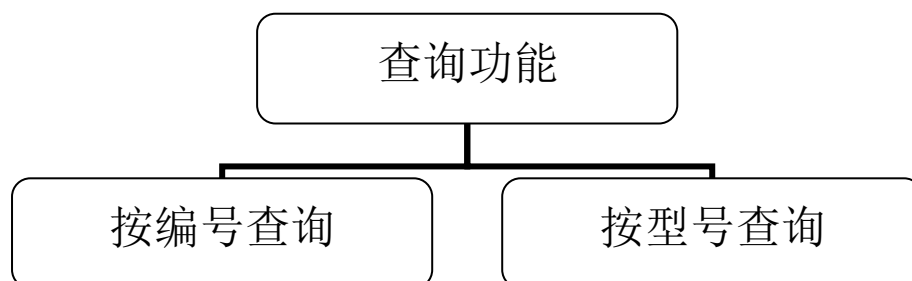


图 5 查询功能模块图

④删除功能模块（按编号删除）

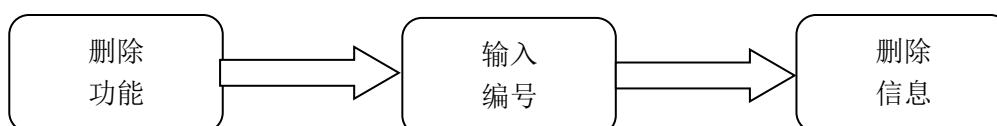
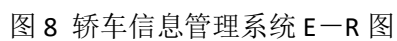


图 6 删除功能模块图

```
graph TD; A[修改功能] --> B[修改型号]; A --> C[修改颜色]; A --> D[修改生产商家]; A --> E[修改生产时间]; A --> F[修改价格];
```

3. 详细设计

数据库概念结构设计 (E-R 图)



数据库逻辑结构设计（数据库表）

表 1 Carinformation 汽车信息表

Number 编号	Model 型号	Colour 颜色	Manufacturer 生产厂家	date of production 出厂日期	价格 Price
1	a	black	a1	2020	256231
2	b	white	b2	2020	654123
3	c	blue	c3	2018	32561
4	d	white	d4	2016	32659
5	e	red	e5	2020	39946
6	f	white	f6	2020	98564
7	g	black	g7	2019	289745
8	h	blue	h8	2016	623541
9	i	black	i9	2015	56214
10	j	white	j10	2020	89564
11	k	red	K11	2017	56231

（2）各模块设计

汽车信息管理系统定义了 sys 类,类中的用户可访问程序接口有 init()初始化函数 function()功能选择函数, secede()系统函数退出, Input()输入信息函数, display()显示信息函数, search()查找信息函数, delete1()删除信息函数 modify()修改信息函数。

①连接与显示功能

Mysql 和 VS 建立连接,在 VS 中新建的 Mysql 结构体 mysql 和 sock,通过 Mysql 所带的函数接口 mysql_init 和 mysql_real_connect 来实现 Mysql 和 VS 的建立连接,从而实现系统类中 init()函数的构建。

汽车信息的输入利用sys类中的Input()输入信息函数,通过while实现多条信息的输入,通过SQL语句" insert into carinformation(number, model ,colour , manufacturer , date , price) values(%d,'%s','%s','%s',%d,%d)", number, model, colour, manufacturer, date, price "和mysql_query函数接口运行SQL语句,成功则输出"输入成功",失败则输出错误语句信息。

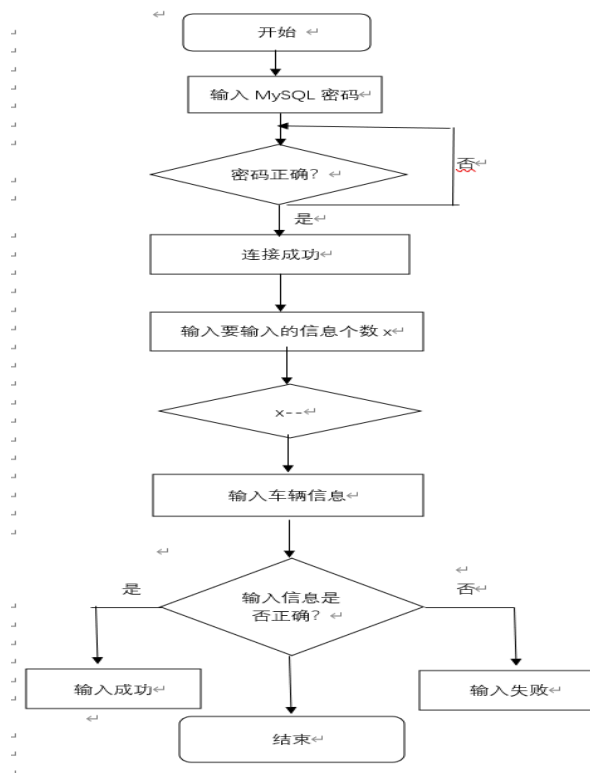


图 9 数据库的连接和输入

②显示功能

显示轿车信息，通过 sys 类中的 display()显示信息函数，通过类中的变量 query_sql 保存 SQL 语句，将"select * from carinformation"SQL 语句通过 sprintf_s 格式化保存在 query_sql 字符串中，通过 mysql_query 函数接口运行 SQL 语句，成功则输出汽车有关信息，失败输出错误。

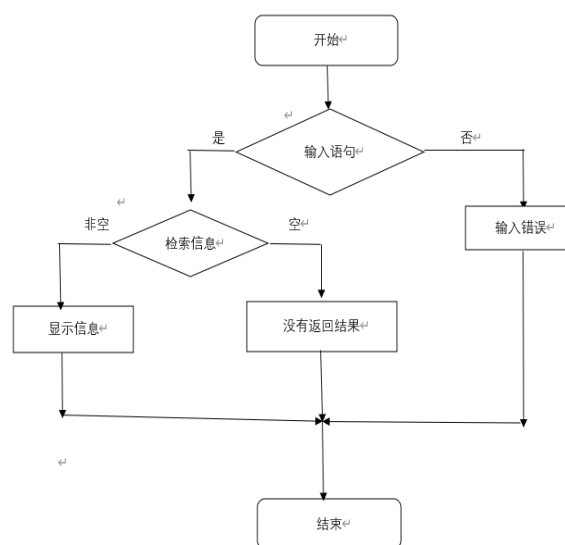


图 10 输入功能流程图

③查找功能

按编号查找

汽车信息的查找利用 sys 类中的 search()查找信息函数,通过 SQL 语句"SELECT * FROM carinformation where number=%d "和 mysql_query 函数接口运行 SQL 语句，成功则输出汽车有关信息，失败输出错误。

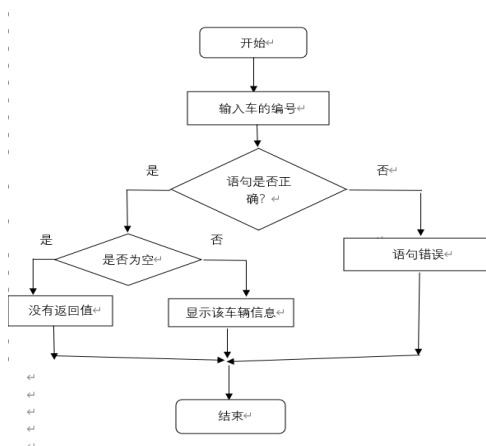


图 11 按编号查找流程图

按型号查找

汽车信息的查找利用 sys 类中的 search()1 查找信息函数,通过 SQL 语句"SELECT * FROM carinformation where model='%s'", model "和 mysql_query 函数接口运行 SQL 语句，成功则输出汽车有关信息，失败输出错误。

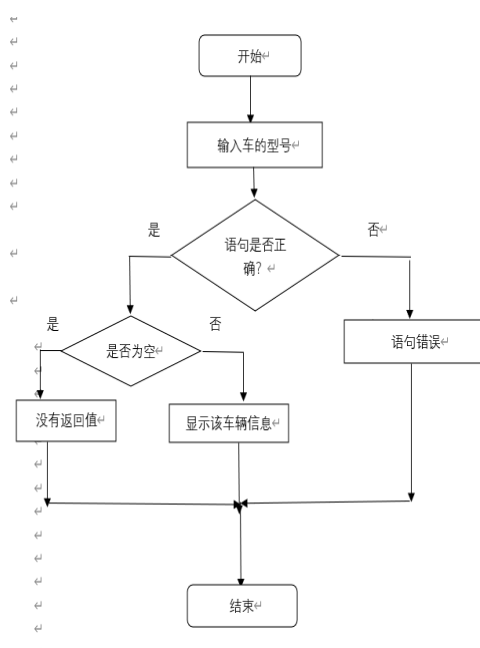


图 12 按型号查找流程图

④删除

汽车信息的删除利用 sys 类中的 delete1()删除信息函数,通过 SQL 语句"DELETE FROM carinformation WHERE number=%d", number"和 mysql_query 函数接口运行 SQL 语句,成功则输出"删除成功",失败则输出错误。

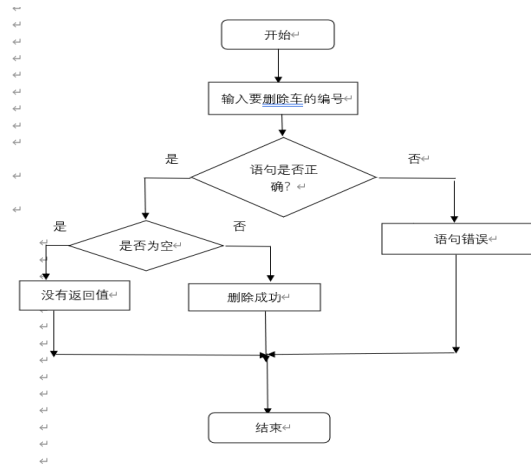


图 13 删除功能流程图

⑤修改

汽车信息的修改利用 sys 类中的 modify()修改信息函数,通过 SQL 语句"update carinformation set Model='%s',Colour='%s',Manufacturer='%s',date=%d,price=%d where Number=%d", model, colour, manufacturer, date, price, number "和 mysql_query 函数接口运行 SQL 语句,成功则输出"修改成功",失败则输出错误。

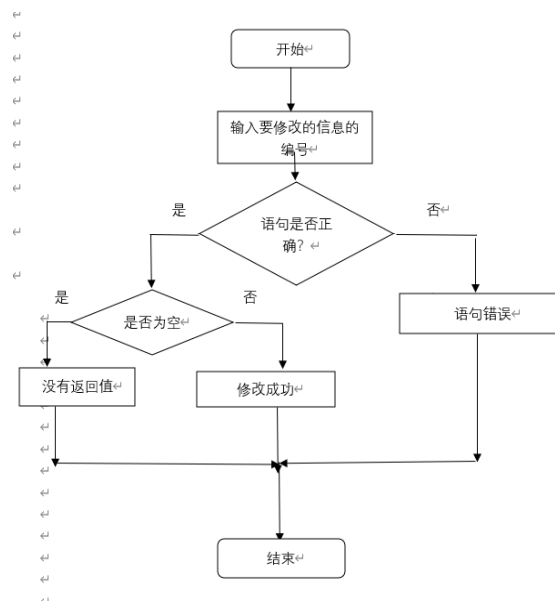


图 14 修改功能流程图

4. 系统实现与调试

(1) 代码

```
#include <stdio.h>

#include<iostream>

#include <stdlib.h>

#include<Windows.h>

#include<mysql.h>

#include<iomanip>

#pragma comment(lib, "libmysql.lib")

using namespace std;

class sys {

public:

    void init();//初始换

    void function();//功能选择

    void secede();//退出

    void Input();//输入

    void display();//显示

    void search();//按编号查找

    void search1();//按型号查找

    void delete1();//删除

    void modify();//修改

private:

    MYSQL mysql, * sock; //声明 mysql 的句柄

    const char* host = "localhost"; //本机连接

    const char* user = "root"; //mysql 用户名

    char passwd[100] = {}; //用户密码

    const char* db = "car"; //汽车信息数据库

    unsigned int port = 3306; //mysql 服务器端口设置在 3306

    const char* unix_socket = NULL; //unix_socket
```



```

unsigned long client_flag = 0; //参数一般为 0
MYSQL_RES* result; //保存结果集
MYSQL_ROW row; //代表的是结果集中的一行
unsigned int num_fields; //结果集某行的列数
char query_sql[800] = {}; //存放 sql 语句
int number; //编号
char model[50] = {}; //型号
char colour[50] = {}; //颜色
char manufacturer[50] = {}; //生产商家
int date; //生产日期
int price; //价格
int stock; //库存
};

void sys::init() //初始化系统
{
    mysql_init(&mysql); //连接之前必须使用这个函数来初始化
    mysql_options(&mysql, MYSQL_SET_CHARSET_NAME, "gbk");
    cout << "链接 mysql, 请输入 mysql 密码" << endl;
    cin.get(passwd, 100);
    if ((sock = mysql_real_connect(&mysql, host, user, passwd, db, port,
unix_socket, client_flag)) == NULL) //连接 MySQL
    {
        cout << "与 mysql 建立连接失败" << endl;
        cout << mysql_error(&mysql) << endl; //输出错误信息
        exit(1); //退出程序
    }
    else
    {
        cout << "与 Mysql 服务器连接成功" << endl;
    }
}

```

```

}

void sys::secede()//退出系统
{
    cout << "退出系统" << endl;
    exit(0);

}

void sys::Input()
{
    cout << "请输入要输入的信息的个数" << endl;
    int x;
    cin >> x ;
    while (x--)
    {
        cout << "请输入汽车的编号、型号、颜色、生产厂家、出厂日期、价
格" << endl;

        cin >> number >> model >> colour >> manufacturer >> date >> price ;
        sprintf_s(query_sql,
            "insert into carinformation(number, model ,colour , manufacturer ,
date , price ) values(%d,'%s','%s','%s',%d,%d)"
            , number, model, colour, manufacturer, date, price);
        if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL
语句,成功返回 0
        {
            cout << "sql 语句出现错误" << endl;
            cout << mysql_error(&mysql) << endl;//sql 语句错误信息
            cout << "输入失败" << endl;
        }
        else
        {

```

```

        cout << "输入成功" << endl;
    }
}

void sys::display()
{
    //显示
    sprintf_s(query_sql, "select * from carinformation");
    if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL 语
句,成功返回 0
    {
        cout << "sql 语句出现错误" << endl;
        cout << mysql_error(&mysql) << endl; //sql 语句错误信息
    }
    else
    {
        result = mysql_store_result(&mysql);
        if (result == NULL) //保存查询的结果
        {
            if (mysql_field_count(&mysql) != 0)
            {
                cout << "没有返回结果" << endl;
            }
        }
        else
        {
            cout << "编号    型号    颜色 生产厂家 出厂日期  价格"
<< endl;

```

while ((row = mysql_fetch_row(result)) != NULL) //读取结果集中的数据

```
{
    num_fields = mysql_num_fields(result);
    for (unsigned i = 0; i < num_fields; i++)
    {
        printf(" %s\t", row[i]);
    } //打印当前行的数据
    printf("\n");
}
```

```
}
```

```
mysql_free_result(result); //释放结果集
```

```
}
```

```
void sys::search()
```

```
{
```

```
    //按编号查找
```

```
    cout << "请输入要查询的车的编号" << endl;
```

```
    cin >> number;
```

```
    sprintf_s(query_sql, "SELECT * FROM carinformation where number=%d",
number);
```

```
    if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL
语句,成功返回 0
```

```
    {
```

```
        cout << "sql 语句出现错误" << endl;
```

```
        cout << mysql_error(&mysql) << endl; //sql 语句错误信息
```

```
    }
```

```
    else
```

```
    {
```

```

result = mysql_store_result(&mysql);
if (result == NULL) //保存查询的结果
{
    if (mysql_field_count(&mysql) == 0)
    {
        cout << "没有返回结果" << endl;
    }
}
else
{
    cout << "编号    型号    颜色 生产厂家 出厂日期    价
格" << endl;

    while ((row = mysql_fetch_row(result)) != NULL) //读取结果
集中的数据

    {
        num_fields = mysql_num_fields(result);
        for (unsigned i = 0; i < num_fields; i++)
        {
            printf(" %s\t", row[i]);
        }//打印当前行的数据
        printf("\n");
    }

}

mysql_free_result(result); //释放结果集
}

void sys::search1()
{

```

```

//按型号查找
cout << "请输入要查询的车的型号" << endl;
cin >> model;
sprintf_s(query_sql, "SELECT * FROM carinformation where model='%s'",
model);
if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL 语
句,成功返回 0
{
    cout << "sql 语句出现错误" << endl;
    cout << mysql_error(&mysql) << endl; //sql 语句错误信息
}
else
{
    result = mysql_store_result(&mysql);
    if (result == NULL) //保存查询的结果
    {
        if (mysql_field_count(&mysql) == 0)
        {
            cout << "没有返回结果" << endl;
        }
    }
    else
    {
        cout << "编号    型号    颜色 生产厂家 出厂日期    价格"
<< endl;

        while ((row = mysql_fetch_row(result)) != NULL) //读取结果集中
的数据
        {
            num_fields = mysql_num_fields(result);
            for (unsigned i = 0; i < num_fields; i++)

```

```

        {
            printf(" %s\t", row[i]);
        }//打印当前行的数据
        printf("\n");
    }
}

}

mysql_free_result(result); //释放结果集
}

void sys::delete1()
{
    cout << "请输入要删除的汽车编号" << endl;
    cin >> number;
    sprintf_s(query_sql,
        "DELETE FROM carinformation WHERE number=%d", number);
    if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL 语
句,成功返回 0
    {
        cout << "sql 语句出现错误" << endl;
        cout << mysql_error(&mysql) << endl;//sql 语句错误信息
        cout << "删除失败" << endl;
    }
    else
    {
        cout << "删除成功" << endl;
    }
}

}

void sys::modify()

```

```

{
    cout << "请输入要修改的信息的编号" << endl;
    cin >> number;
    cout << "请分别输入要修改的型号、颜色、生产厂家、出厂日期、价格"
<< endl;
    cin >> model >> colour >> manufacturer >> date >> price ;
    sprintf_s(query_sql,
               "update                               carinformation                               set
Model='%s',Colour='%s',Manufacturer='%s',date=%d, price=%d where Number=%d"
               , model, colour, manufacturer, date, price, number);
    if (mysql_query(&mysql, query_sql) != 0) //mysql_query 函数运行 SQL 语
句,成功返回 0
    {
        cout << "sql 语句出现错误" << endl;
        cout << mysql_error(&mysql) << endl;//sql 语句错误信息
        cout << "修改信息失败" << endl;
    }
    else
    {
        cout << "修改信息成功" << endl;
    }
}

void sys::function()
{
    cout << "进入汽车管理系统" << endl << endl;
    cout << "1.输入信息" << endl;
    cout << "2.显示信息" << endl;
    cout << "3.按编号查找信息" << endl;
    cout << "4.按型号查找信息" << endl;
}

```



```

    cout << "5.删除信息" << endl;
    cout << "6.修改信息" << endl;
    cout << "7.退出" << endl;
}
int main()
{
    int function_ch;
    char if_ch;
    sys carsys;
    carsys.init();
    carsys.function();
    while (1)
    {
        cout << endl;
        cout << "请输入你要进行的操作" << endl;
        cin >> function_ch;
        while (function_ch > 7 || function_ch < 1)
        {
            cout << "输入错误，请重新输入" << endl;
            cin >> function_ch;
        }
        switch (function_ch)
        {
            case 1: carsys.Input(); break;
            case 2: carsys.display(); break;
            case 3: carsys.search(); break;
            case 4: carsys.search1(); break;
            case 5: carsys.delete1(); break;
            case 6: carsys.modify(); break;
            case 7: carsys.secede();

```

```

    }

    cout << endl << "如果你想继续,请输入 E,否则退出系统" << endl;

    cin >> if_ch;

    if (if_ch != 'E')

        break;

    }

    cout << "成功退出系统" << endl;

    exit(0);

}

```

（2）出现的问题

定义 sys 类，类中有 init()初始化函数 function()功能选择函数，secede()系统函数退出，Input()输入信息函数，display()显示信息函数，search()查找信息函数，delete1()删除信息函数 modify()修改信息函数，在设置这些函数的是时候出现了语法错误。SQL 语句与 mysql_query 函数接口运行 SQL 语句时出现错误。

（3）解决方法

运行代码，通过 VS 中解决方案中错误代码出现的行数，按照行数找到代码错误的地方对语法进行修改。对于 SQL 语句与 mysql_query 函数接口运行 SQL 语句时出现的错误和同学讨论后解决了。

5. 系统使用测试说明

（1）运行程序输入数据库密码，选择功能前的数字



(2) 若要进行输入功能输入 1, 然后输入你要输入信息个数和输入的信息。

```
D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
链接mysql, 请输入mysql密码
100861
与Mysql服务器连接成功
进入汽车管理系统

1. 输入信息
2. 显示信息
3. 按编号查找信息
4. 按型号查找信息
5. 删除信息
6. 修改信息
7. 退出

请输入你要进行的操作
1
请输入要输入的信息的个数
2
请输入汽车的编号、型号、颜色、生产厂家、出厂日期、价格
14 n red n14 2020 159687
输入成功
请输入汽车的编号、型号、颜色、生产厂家、出厂日期、价格
15 o red o15 2018 159658
输入成功

如果你想继续, 请输入E, 否则退出系统
```

(3) 若继续查看输入的结果则进行第二个功能, 输入 E, 再输入 2。

```
D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
如果你想继续, 请输入E, 否则退出系统
E

请输入你要进行的操作
2

编号    型号    颜色  生产厂家  出厂日期  价格
1        a      black a1        2020     256231
2        b      white b2        2020     654123
3        c      blue  c3        2018     32561
4        d      white d4        2016     32659
5        e      red   e5        2020     39946
6        f      white f6        2020     98564
7        g      black g7        2019     289745
8        h      blue  h8        2016     623541
9        j      white i9        2015     56214
10       i      white j10       2020     89564
11       k      red   k11       2017     56231
12       l      red   l12       2020     888888
13       m      red   m13       2018     888888
14       n      red   n14       2020     159687
15       o      red   o15       2018     159658

如果你想继续, 请输入E, 否则退出系统
```

(4) 若继续要对某个编号的汽车信息进行查询, 输入 E 后再输入 3, 最后输入你要查找车的编号。

```
D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
如果你想继续, 请输入E, 否则退出系统
E

请输入你要进行的操作
3
请输入要查询的车的编号
5

编号    型号    颜色  生产厂家  出厂日期  价格
5        e      red   e5        2020     39946

如果你想继续, 请输入E, 否则退出系统
```

(5) 若继续要对某个型号的汽车信息进行查询，输入 E 后再输入 4, 最后输入你要查找车的型号。

```

D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
如果你想继续, 请输入E, 否则退出系统
E
请输入你要进行的操作
4
请输入要查询的车的型号
f
编号    型号    颜色 生产厂家 出厂日期  价格
6        f        white f6      2020    98564
如果你想继续, 请输入E, 否则退出系统

```

(6) 若继续要对某个编号的汽车信息进行删除，输入 E 后再输入 5, 最后输入你要删除车的编号，若要查看删除结果重复 (2)。

```

D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
如果你想继续, 请输入E, 否则退出系统
E
请输入你要进行的操作
5
请输入要删除的汽车编号
11
删除成功
如果你想继续, 请输入E, 否则退出系统
E
请输入你要进行的操作
2
编号    型号    颜色 生产厂家 出厂日期  价格
1        a        black a1      2020    256231
2        b        white b2      2020    654123
3        c        blue  c3      2018    32561
4        d        white d4      2016    32659
5        e        red   e5      2020    39946
6        f        white f6      2020    98564
7        g        black g7      2019    289745
8        h        blue  h8      2016    623541
9        j        white i9      2015    56214
10       i        white j10     2020    89564
12       l        red   112     2020    888888
如果你想继续, 请输入E, 否则退出系统

```

(7) 若继续要对某个编号的汽车信息进行修改，输入 E 后再输入 6, 最后输入你要修改车的编号，在输入要修改的信息，若要查看修改结果重复 (2)。

```
D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe
如果你想继续, 请输入E, 否则退出系统
E

请输入你要进行的操作
6
请输入要修改的信息的编号
11
请分别输入要修改的型号、颜色、生产厂家、出厂日期、价格
k blue k11 2020 590000
修改信息成功

如果你想继续, 请输入E, 否则退出系统
E

请输入你要进行的操作
2
编号    型号    颜色  生产厂家  出厂日期  价格
1       a      black a1       2020     256231
2       b      white b2       2020     654123
3       c      blue  c3       2018     32561
4       d      white d4       2016     32659
5       e      red   e5       2020     39946
6       f      white f6       2020     98564
7       g      black g7       2019     289745
8       h      blue  h8       2016     623541
9       j      white i9       2015     56214
10      i      white j10      2020     89564
11      k      blue  k11      2020     590000
12      l      red   l12      2020     888888
14      n      red   n14      2020     159687
15      o      red   o15      2018     159658
```

(8) 若退出系统输入除 E 以外的任意一个则退出系统

```
Microsoft Visual Studio 调试控制台

如果你想继续, 请输入E, 否则退出系统
23
成功退出系统

D:\桌面\程序实习\Git\轿车信息管理系统\轿车信息管理系统\Debug\2.0.exe (
进程 142048) 已退出, 代码为 0。
要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调
试停止时自动关闭控制台”。
按任意键关闭此窗口. . .
```

6. 总结

(1) 系统性能优缺点分析

优点：快速准确的由计算机来代替人工执行一系列诸如输入、显示、查找、删除、修改等操作，非常便捷。

缺点：不能按照颜色、生产厂家、出厂日期、价格对汽车信息进行查修、修改删除；进行输入、删除、修改等操作时只提醒操作成功不能显示 输入、删除、修改的结果。

(2) 经验总结、体会与收获

通过这次为期两个多星期的程序设计训练, 不仅让我对 c 语言的一些知识得到了回顾, 加深了对 c 语言的掌握。同时也让我对数据结构、数据库、面向对象程序设计知识得到了一定的掌握。知道了怎么去分析一个题目, 怎样选择比较好

的数据结构怎样定义一个合适的栈和建立一个合适的类,以及怎样在类中定义函数;也知道了怎么样实现程序功能,转变到怎么用更少的空间,更短的时间完成程序设计。

另外,综合部分的轿车信息管理系统的设计让我对数据库的具体应用建立起了一定的概念,使我深刻意识到数据库的重要性。通过构建自己的数据库,数据表,利用 VS 与数据库连接,再通过 VS 利用 SQL 语句对轿车信息管理系统进行输入、显示、查询、删除、修改等功能,对数据库和面向对象程序设计都有了更加深入的认识;同时,也大大提高了我的编程能力与纠错能力,改进了自主学习的方法。

虽然这次程序设计训练结束了,但学习是永无止境的。要想做出更好的设计仍需付出更多的努力。以后我会继续这一方面的学习,丰富自己的知识体系,做出更完美的程序设计。

参考文献

- [1] 董正言. 面向对象程序设计[M]. 清华大学出版社, 2010.
- [2] 宋浩远. 面向对象程序设计方法概述[J]. 重庆科技学院学报(自然科学版), 2008(02):99-102.
- [3] 范剑波. 网络数据库技术及应用[M]. 西安电子科技大学出版社, 2004.
- [4] 王鹏, 董群. 数据库技术及其应用[M]. 人民邮电出版社, 2000.
- [5] 沈姝. NoSQL 数据库技术及其应用研究[D]. 南京信息工程大学.
- [6] 柳广凤, 徐爱平. 面向轿车整车开发的数据管理系统研究[J]. 黑龙江科技信息, 2008, 000(033):55-55.
- [7] 李汉辉. B 公司汽车营销管理信息系统建设问题的分析与研究[D]. 天津大学.

实习成绩：				
设计实现过程（50）	实习报告（20）	答辩（30）	成绩（分数）	成绩（等级）

评定教师：

评定日期：2021 年 1 月 日