燕山大學
YANSHAN UNIVERSITY

C++面向对象程序设计

实验指导书

（5） 数组、指针与字符串

燕山大学软件工程系

目 录

实验 05 数组、指针与字符串

1.1 时间安排 4 学时

本实验安排 4 个实验课时。

1.2 实验目的和要求

1．学习使用数组。

2．掌握指针的使用方法，体会运算符& 、*的不同作用。

3. 学习字符串数据的组织和处理。

4. 练习通过动态分配内存实现动态数组，并体会指针在其中的作用。

5．分别使用字符数组和标准 C++库练习处理字符串的方法。

1.3 实验内容 I (调试、理解、体会、掌握)

(1) 编写并测试 3×3 矩阵转置函数，使用3×3 数组保存矩阵。

```
#include <iostream>
using namespace std;
void main()
{
    int a[3][3]={1,2,3,4,5,6,7,8,9};
    int i,j,t;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            cout<<a[i][j]<<"   ";
        cout<<endl;
    }
    for(i=0;i<3;i++)
        for(j=i;j<3;j++)
            t=a[i][j],a[i][j]=a[j][i],a[j][i]=t;
    cout<<endl;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
            cout<<a[i][j]<<"   ";
        cout<<endl;
    }
}
```

(2) 建立一个对象数组， 内放 5 个学生的数据(学号、成绩)，用指针指向数组首元素， 输出第 1,3,5 个学生的数据。

```
#include<iostream>
```

```
using namespace std;
class student
{
    private:
        int num;
        int score;
    public:
        void display();
        student(int n,int s):num(n),score(s){}
};
void student::display()
{
    cout<<"student "<<num<<"'s score is "<<score<<endl;
}
int main()
{
    student stu[5]={student(1,91),student(2,92),student(3,93),student(4,94),student(5,95)};
    student *p=stu;
    int i;
    for(i=0;i<3;p=p+2,i++)
    {
        p->display();
    }
    return 0;
      }
```

（3）运行并阅读下列程序，仔细体会取地址运算符”&”和取内容运算符”*”的作用。

```
#include <iostream>
using namespace std;
void main()
{
    int i=100;
    int *ptr=&i;
    cout<<"ptr:"<<ptr<<endl;
    cout<<"*ptr:"<<*ptr<<endl;
    cout<<"&ptr:"<<&ptr<<endl;
}
```

（4）运行并阅读下列程序，仔细体会常量指针与指针常量的区别。

```
#include <iostream>
using namespace std;
void main(){
```

```
// 常量指针  不可以改变指针所指对象的值；但可以能改变对象
    int m=50;
    int n=210;
    const int * ptr1=&m;
    cout<<"*ptr1:"<<*ptr1<<endl;
     //*ptr1=200;  //错误
    ptr1=&n;
    cout<<"*ptr1:"<<*ptr1<<endl;
// 指针类型常量  可以改变指针所指对象的值；但不能改变对象
   int * const ptr2=&m;
    cout<<"*ptr2:"<<*ptr2<<endl;
    *ptr2=200;
    // ptr2=&n; //错误
    cout<<"*ptr2:"<<*ptr2<<endl;
}
```

（5）运行并阅读下列程序，仔细体会利用指针、数组名对数组元素操作的本质不同。

```
#include <iostream>
using namespace std;
int main() {
    cout<<"****二维数组的种指针表示*****"<<endl;
    int i, j, a[2][3] = { { 1, 3, 5 }, { 2, 4, 6 } }, *p;
    p=&a[0][0];
      for (i = 0; i < 2; i++) {
            for (j = 0; j < 3; j++) {
                cout<<a[i][j]<<"    ";
                cout<< *(a[i] + j)<<"    ";
                cout<< *(*(a + i) + j)<<"    ";
                cout<<*(p + 3 * i + j)<<"    ";
            }
          cout<<endl;
    }
    cout<<"**********指针数组**********"<<endl;
    int *pArray[3];
    int Line1[3],Line2[3],Line3[3];
    pArray[0] = Line1; pArray[1] = Line2; pArray[2] = Line3;
    cout<<"****输入二维数组元素*****"<<endl;
    for (int m = 0; m < 3; m++) {
        for (int n=0;n<3;n++){
            cout<<"输入pArray["<<m<<"]["<<n<<"]的值：  ";
            //该形式与数组下标的访问形式相似，但本质不同首地址的来源不同
            cin>>pArray[m][n];


}
cout<<endl;
```

```
}
cout<<"****输出二维数组元素*****"<<endl;
for (int m = 0; m < 3; m++)
{
   for (int n=0;n<3;n++)
   {
      cout<<*(pArray[m]+n)<<" "; //cout<<pArray[m][n]<<" "
   }
cout<<endl;
}
}
```

1.5 实验内容 II (自主完成)

（1）编写 C++程序完成以下功能：

用类来实现矩阵，定义一个矩阵的类，属性包括：

矩阵大小，用 lines, rows（行、列来表示）；

存贮矩阵的数组指针，根据矩阵大小动态申请（new）。

矩阵类的方法包括：

构造函数：参数是矩阵大小，需要动态申请存贮矩阵的数组；

析构函数：需要释放矩阵的数组指针；

拷贝构造函数：需要申请和复制数组（深复制）；

输入函数：可以从 cin 中输入矩阵元素；

输出函数：将矩阵格式化输出到 cout；

矩阵相加函数：实现两个矩阵相加的功能，结果保存在另一个矩阵里，但必须矩阵大小相同；

矩阵相减的函数：实现两个矩阵相减的功能，结果保存在另一个矩阵里，但必须矩阵大小相同。

主函数功能：

定义三个矩阵：A1、A2、A3；

初始化 A1、A2；

计算并输出 A3 = A1+A2，A3=A1+A2；

用 new 动态创建三个矩阵类的对象：pA1、pA1、pA3；

初始化 pA1、pA2；

计算并输出 pA3=pA1+pA2，pA3=pA1-pA2；

释放 pA1、pA1、pA3。

```
/*动态申请二维数组*/
 int** array = new int*[lines];/*行*/
  for(int i=0; i<lines; ++i) {
    array[i] = new int[rows];/*列*/
  }
/*释放内存空间*/
 for(int i=0; i<lines; i++) {
  delete[] array[i];
 }
 delete[] array;
```