

《面向对象程序设计实践（C++）》课程基础实验

1、C++基础知识实验

编写C++程序完成“矩阵”以下功能：

- (1) 假定矩阵大小为 4×5 （整型）；
- (2) 矩阵空间采用 `new` 动态申请，保存在指针中；
- (3) 定义矩阵初始化函数，可以从 `cin` 中输入矩阵元素；
- (4) 定义矩阵输出函数，将矩阵格式化输出到 `cout`；
- (5) 定义矩阵相加的函数，实现两个矩阵相加的功能，结果保存在另一个矩阵中；
- (6) 定义矩阵相减的函数，实现两个矩阵相减的功能，结果保存在另一个矩阵中；
- (7) 动态申请三个矩阵：A1、A2、A3；
- (8) 初始化 A1、A2；
- (9) 计算并输出 $A3 = A1 + A2$ ， $A3 = A1 - A2$ ；
- (10) 释放矩阵空间。

2、类与对象实验

2.1 编写C++程序完成“圆形”以下功能：

- (1) 定义一个 `Point` 类，其属性包括点的坐标，提供计算两点之间距离的方法；
（要求当用户不输入坐标数据时，能够默认为坐标原点（0，0））
- (2) 定义一个圆形类，其属性包括圆心和半径；
- (3) 创建两个圆形对象，提示用户输入圆心坐标和半径，判断两个圆是否相交，并输出结果；
- (4) 观察圆形对象以及 `Point` 类成员的构造函数与析构函数的调用。
（提示及要求：1，可通过在构造与析构函数中加入输出提示信息的方式观察相关调用；可以使用 `system("pause")` 进行程序的暂停；2，能够理解并说明每一次构造与析构函数调用是哪个对象的调用，并观察和解释相关调用顺序及其原因）

2.2 编写C++程序完成“矩阵”类以下功能：

- (1) 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`）。
- (2) 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；
 - 输入，可以从 `cin` 中输入矩阵元素；
 - 输出，将矩阵格式化输出到 `cout`；
 - 矩阵相加的函数，实现两个矩阵相加的功能，结果保存在另一个矩阵类，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类，但必须矩阵大小相同。
- (3) 定义三个矩阵：A1、A2、A3；
- (4) 初始化 A1、A2；

- (5) 计算并输出 $A3 = A1$ 加 $A2$, $A3 = A1$ 减 $A2$; (要求及提示: 最好能实现对赋值操作符 “=” 的重载; 注意检查 “自赋值”、释放 “旧元素”)
- (6) 用 new 动态创建三个矩阵类的对象: pA1、pA1、pA3;
- (7) 初始化 pA1、pA2;
- (8) 计算并输出 pA3=pA1 加 pA2, pA3=pA1 减 pA2;
- (9) 释放 pA1、pA2、pA3。

3、继承与派生实验

编写C++程序完成“形状”的以下功能:

- (1) 声明一个基类 Shape (形状), 其中包含一个方法来计算面积;
- (2) 从 Shape 派生两个类: 矩形类和圆形类;
- (3) 从矩形类派生正方形类;
- (4) 分别实现派生类构造函数、析构函数和其他方法;
- (5) 创建派生类的对象, 观察构造函数、析构函数调用次序;
(提示及要求: 1, 可通过在构造与析构函数中加入输出提示信息的方式观察相关调用; 可以使用 system("pause")进行程序的暂停; 2, 能够理解并说明每一次构造与析构函数调用是哪个对象的调用, 并观察和解释相关调用顺序及其原因)
- (6) 对不同对象计算面积。

4、I/O 流实验

编写 C++程序完成猜价格游戏的以下功能:

- (1) 假定有一件商品, 程序用随机数指定该商品的价格 (1-1000 的整数);
- (2) 提示用户猜价格, 并输入: 若用户猜的价格比商品价格高或低, 对用户作出相应的提示;
- (3) 直到猜对为止, 并给出提示。
(提示及要求: 1, 要求使用 C++的输入输出方式 (cin, cout), 不能使用 C 语言的 printf 等; 2, 注意检查输入的合法性)

5、重载实验

5.1 虚函数

针对题目3的“形状”类, 编写C++程序完成以下功能:

- (1) 将【形状】中的基类计算面积的方法定义为虚函数, 比较与【形状(A)】程序的差异;
- (2) 将【形状】中的基类定义为抽象类, 比较与【形状(A)】程序的差异。

5.2 对Point类重载++和--运算符

编写C++程序完成以下功能:

- (1) Point 类的属性包括点的坐标 (x, y);
- (2) 实现 Point 类重载++和--运算符:
 - ++p, --p, p++, p--;
 - ++和--分别表示 x, y 增加或减少 1。