

# 实验六 继承与派生

## 1 实验目的

学习继承与派生的相关理论，熟悉不同继承方式下对基类成员的访问方式，包括以下几个方面：

- (1) 学习声明和使用类的继承关系，声明派生类；
- (2) 熟悉不同继承方式下，对基类成员的访问控制；

## 2 实验内容

### 2.1 停车场程序

#### (1) 问题描述

请根据题目要求完成简单的停车场管理程序。

1. 停车场 (Park) 有 N 个停车位 (space)，每个停车位可以停放不同类型的汽车 (Automobile)，包括卡车 (Truck)、轿车 (Car)、公交车 (Bus)，但同一时刻一个停车位只能停放 0 或 1 辆汽车。如果没有空余停车位，显示提示信息，但不会为车辆安排停车位。

2. 程序模拟车辆停车的情况：新来车辆时如果有空位，按顺序为该车分配停车位；车辆开走时，应交纳停车费。

3. 停车场可以显示当前停放的车辆的车牌号码，以及当前的全部停车费收入 (income)。

4. 定义汽车基类 Automobile，包括车牌号码 (字符串) 成员数据。

5. 定义派生类 Truck、Car、Bus。这些车辆除了拥有车牌号码之外，还各自拥有不同的属性。Truck 还包括载重量属性 (浮点数，单位吨)；Car 还拥有品牌属性 (字符串)，Bus 还包括核定载员数量 (整型)。

此外，每个派生类中要实现 pay() 函数，用于显示车辆信息并交纳停车费。

其中, Truck 收费 3 元/次, Car 收费 1 元/次, Bus 收费 2 元/次。

## (2) 问题要求

编写程序, 测试上述所要求的各种功能。要求创建新的工程项目 ParkManager, 添加必要的源文件和头文件, 并在程序适当的位置中编写注释。

```
class Automobile {};    // 汽车类
class Park {};          // 停车场类

int main() {
    cout << "请输入停车位数量: ";
    cin >> N; // 输入停车位数量, 此处输入2

    Park park(N); // 创建一个停车场对象

    Car car1("鲁B-12345", "奥迪A6"); // 创建轿车对象
    car1.enter(park); // car1进入停车场, 分配停车位

    Truck truck("鲁B-23456", 15); // 创建卡车对象
    truck.enter(park); // truck进入停车场, 分配车位

    car1.leave(park); // car1离开停车场, 缴纳停车费

    Bus bus("鲁B-34567", 50); // 创建公交车对象
    bus.enter(park); // bus进入停车场, 分配车位

    /* 显示当前停放的车辆的车牌号码, 以及当前的全部停车费收入 */
    park.showInfo();

    Car car2("鲁B-45678", "宝马320"); // 创建轿车对象
    car2.enter(park);
    // car2进入停车场, 分配停车位。因为没有空余停车位, 所以无法分配

    bus.leave(park); // bus离开停车场, 缴纳停车费
    truck.leave(park); // truck离开停车场, 缴纳停车费
```

```

/* 显示当前停放的车辆的车牌号码，以及当前的全部停车费收入*/
park.showInfo();

return 0;
}

```

## (2) 程序执行结果

程序执行结果如下：

请输入停车位数量：2

鲁B-12345进入停车场，分配停车位

鲁B-23456进入停车场，分配停车位

鲁B-12345离开停车场，缴纳停车费1元

鲁B-34567进入停车场，分配停车位

停车场目前停放了2辆汽车：鲁B-23456，鲁B-34567，共收入1元停车费

无法为鲁B-45678分配停车位

鲁B-34567离开停车场，缴纳停车费2元

鲁B-23456离开停车场，缴纳停车费3元

停车场目前停放了0辆汽车，共收入6元停车费

提示：停车场的停车位不能再用布尔型数组表示，要用Automobile的指针数组表示。

```

Automobile **spaces;
spaces = new Automobile*[N];
spaces[i] = &car1;

delete[] spaces;

```