**第4章练习三**

1. **定义一个Vehicle类**，包含成员变量：float x\_, float y\_, float vx \_, float vy\_, float width\_, float height\_以及static int vCount\_，表示已经创建的车辆对象个数。

其中：(x\_, y\_)表示车辆中心的初始位置坐标，单位m，(vx\_,vy\_)表示车辆的初始速度，单位m/s，width\_和height\_分别表示车辆的宽和长，单位m；

* 实现Vehicle类的构造函数、复制构造函数、析构函数；

其中：vx\_和vy\_的默认值分别为：6，0；width和height的默认值分别为2，4

* 实现void Display()函数，显式车辆基本信息；
* 实现成员函数IsCollision(const Vehcile &veh)：判断自身是否与车辆veh发生了碰撞，如果发生碰撞，则返回1，否则返回0。

1. **定义一个LaneSegment类**，包含成员变量：float x\_, float y\_, float length\_, float width\_。

其中：(x\_, y\_)表示车道左上角的起始位置，length\_分别表示车道和长度，单位m；Vehicle对象数组；static const int maxVehicles表示最大车辆数。

* 实现构造函数、复制构造函数和析构函数；

其中：length\_和width\_的默认值分别为：100，3.5

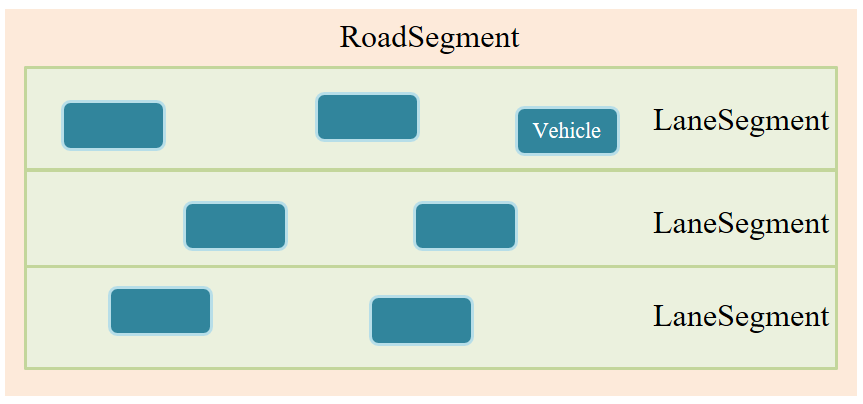
* 实现成员函数IsInLane(const Vehicle &veh)：判断车道上是否存在车辆veh，如果存在返回1，否则返回0；
* 实现成员函数AddVehicle((Vehicle &veh)：在道路上生成新的车辆veh；
* 实现成员函数RemoveVehicle(Vehicle &veh)：在道路上删除车辆veh。

1. **定义一个RoadSegment类**，包含：

* 实现构造函数和析构函数；
* 设置static const int maxLanes成员，表示最大车道数；
* 构建Lane对象数组；
* 实现成员函数AddLane(Lane &lane)：在道路中增加车道lane。

**实现非成员函数：**初始化场景函数InitScenario()

**关系示意图：**



**实验步骤：**

1. 新建RoadSegment，自下而上生成三个车道
2. 通过随机函数在每个车道中生成至少2辆车
3. 生成过程中，如果车辆发生了碰撞，则重新生成，直至不发生碰撞
4. 删除最下面的车道及其车道内的车辆

文件组织：每一个类都单独实现为h文件和cpp文件，在main函数内部进行测试。