

# Week7

## 1. 定义一个 Point 类，包含：

- 两个int型数据成员 x, y，表示点的横纵坐标（私有成员）
- 一个无参构造函数，默认点坐标为原点
- 一个有参构造函数，一个用给定的x和y作为参数创建一个点，用this指针
- 设置函数与获取函数（**Set, Get**）
- 欧式距离函数：计算两点之间的欧式距离
- 输出函数，输入为一个点，输出两个点之间的欧式距离，输出格式为：(x1,y1)\*(x2,y2) = z（保留两位小数）

## 2. 定义一个 Rectangle 类，该类表示平面上的一个矩形，要求：

- 使用 Point 类的对象来描述 Rectangle 的两个顶点坐标(左上角、右下角)
- 一个计算矩形周长的函数
- 一个计算矩形面积的函数
- 一个输出函数，可以输出矩形的四个顶点(左上顶点开始，顺时针方向)、周长及面积

## 3. 在 Point 和 Rectangle 的基础上，实现一个矩形链表类，在该链表类中，矩形以面积升序（从小到大）的形式存储，包含：

- 构造函数、析构函数
- append 函数，函数接受一个矩形作为参数，根据矩形面积大小将其插入到链表对应位置中（如果面积大小相等，根据周长进行排序）
- pop 函数，接受一个 int 型参数i，返回链表中第 i 个矩形，并从链表中删除该矩形
- 一个输出函数，输出当前链表面积最大的三个矩形的信息（顶点、周长及面积）

输出：

```
1 1
2 vertices: (x1, y1) (x2, y2) (x3, y3) (x4, y4)
3 perimeter: 18
4 area: 14
5
6 2
7 vertices: (x1, y1) (x2, y2) (x3, y3) (x4, y4)
8 perimeter: 16
9 area: 12
10
11 3
12 vertices: (x1, y1) (x2, y2) (x3, y3) (x4, y4)
13 perimeter: 14
14 area: 10
```