

它的值。

的确，这个讨论看起来可能有些“抽象”（抱歉！），不用担心，只需掌握基本的概念就可以。

8.5 练习

1. 向代码清单 8-1 添加一个名为 ClassC 的新类，它是 ClassB 的子类。创建一个 initVar 方法，它将实例变量 x 的值设置为 300。编写一个测试方法，它声明对象 ClassA、ClassB 及 ClassC，并且调用相应的 initVar 方法。
2. 使用高分辨率的设备时，可能需要使用允许将点指定为浮点值，而不是简单的整数（iOS 中使用 CGRect 结构用在矩形上，矩形所有的坐标和尺寸均用浮点数表示）坐标系。修改本章的 XYPoint 与 Rectangle 类，以处理浮点数字。矩形的宽度、高度、面积与周长也都使用浮点数字进行处理。
3. 修改代码清单 8-1，向其添一个名为 ClassB2 的新类，ClassB2 和 ClassB 一样，都是 ClassA 的子类。ClassB 与 ClassB2 之间有什么关系？指出 NSObject 类、ClassA、ClassB 及 ClassB2 之间的层次关系。ClassB 的超类是什么？ClassB2 的超类是什么？一个类可以有多少个子类、多少个超类？
4. 编写一个名为 translate: 的 Rectangle 方法，使用 XYPoint 对象作为其参数。通过指定的向量对矩形的原点进行变换。注意，变换是指一个点移动到另一个点。
5. 定义一个名为 GraphicObject 的新类，使其成为 NSObject 的子类。在新类中定义如下一些实例变量：

```
int fillColor;    // 32 位颜色
BOOL filled;      // 是否为对象填充了？
int lineColor;    // 32 位线的颜色
```

编写一个方法，设定并检索前面定义的变量。使 `Rectangle` 类成为 `GraphicObject` 的子类。

定义两个新类 `Circle` 和 `Triangle`，它们都是 `GraphicObject` 的子类。编写一些方法来设定及检索这些对象的各种参数，并计算圆的圆周、面积及三角形的周长、面积。

6. 编写一个名为 `containsPoint:` 的 `Rectangle` 方法，使用 `XYPoint` 对象作为参数

```
-(BOOL) containsPoint: (XYPoint *) aPoint;
```

这个方法返回 `BOOL` 值，如果矩形包含有这个点，返回 `YES`，否则返回 `NO`。

7. 编写一个名为 `intersect:` 的 `Rectangle` 方法，该方法使用一个矩形作为参数，并返回代表两个矩形的重叠区域。例如，给定图 8.10 所展示的两个矩形，这个方法应该返回原点位于 (400, 420) 的矩形，其宽度为 50、高度为 60。

如果矩形没有相交，返回宽度与高度均为零的矩形，其原点为 (0, 0)。

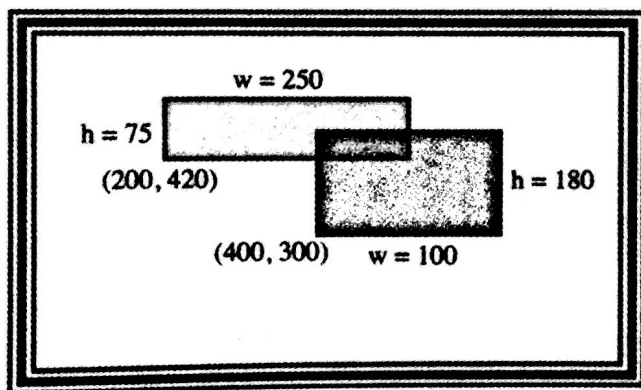


图 8.10 相交的矩形

8. 为 `Rectangle` 类编写一个名为 `draw` 的方法，此方法使用虚线与垂直的条形字符绘制矩形。以下代码序列

```
Rectangle *myRect = [[Rectangle alloc] init];  
[myRect setWidth: 10 andHeight: 3];  
[myRect draw];
```

将产生如下输出结果：

```
-----  
|       |
```

```
|      |  
|      |  
-----
```

注意

应该使用 `printf` 绘制字符，因为每次调用 `NSLog` 时都会显示一个新行。