NJUSWI
2002-2012
Anniversary
南京大學 软件学院

类和对象

刘 钦

南京大学软件学院

什么是对象

- 一种建模方法
  - 既表示客观世界问题空间(Namespace)中的某个具体的事物，
  - 又表示软件系统解空间中的基本元素。
- 对象包括
  - 属性(Properties)
    - 属性就是需要记忆的信息
  - 方法(Methods)
    - 方法就是对象能够提供的服务。

什么是对象

- 每个对象都保存着描述当前特征的信息。
- 对象状态的改变必须通过调用方法实现。
- 对象的状态不能完全描述一个对象。每个对象都有一个唯一的身份(identity)。
- 每个对象的标识永远是不同的，状态常常也存在着差异。

如何获得对象

- 寻找候选对象
  - 找名词 -- 类（对象）与属性
  - 找动词 -- 行为
- 精化对象
  - 去除
    - 冗余（VIP等价于会员）
    - 不相干（超市收银系统中的顾客）
    - 模糊的概念（xx信息、xx特征）
- 转化
  - 没有行为的对象-》某个类的属性

什么是类？

- 描述相同事物的集合。
- 它以概要的方式描述了相同事物集合中的所有元素，但却允许类中的每一个实体元素可以在非本质特征上变化。

什么是类？

- 类（Class）这个术语是对具有共同具体属性的对象的描述。
- 类是一个描述或蓝图（被表示成一段代码），用于定义组成某类特定对象的所有特性。
- 编程中使用类的思想与现实世界中把东西进行分类的思想相一致，这是一种方便而明确的事物组织方式。

什么是类？

- 一旦定义了一个类，就可以接着得到这个类的对象或实例。
- 实例变量的值由类的每个实例提供。
- 当创建类的实例时， 就建立了这种类型的一个对象，然后系统为类定义的实例变量分配内存。

类与对象

- 类是对某个对象的定义。
- 它包含有关对象动作方式的信息，包括它的名称、方法、属性和事件。
- 当引用类的代码运行时，类的一个新的实例，即对象，就在内存中创建了。虽然只有一个类，但能从这个类在内存中创建多个相同类型的对象。

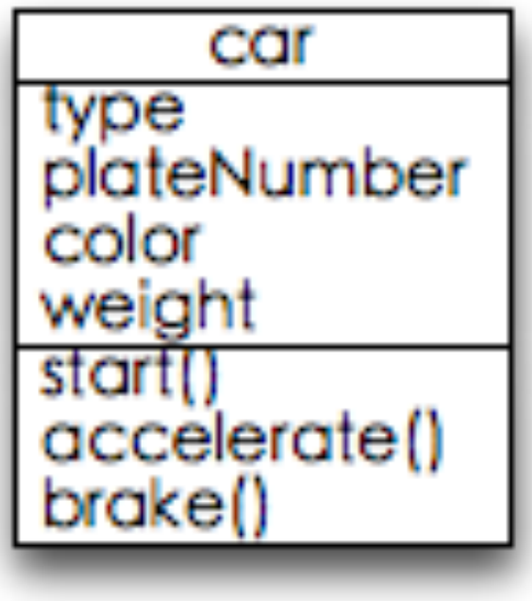
创建类的原因

- 对现实世界中的对象建模
  - 学生、老师
- 对抽象对象建模
  - 形状，事件，事物
- 隐藏实现细节
  - xx营销策略
- 创建中心控制点
  - 控制器
- 让参数传递更顺畅
  - PO
- 隔离复杂度
  - 工厂

类的定义

- class MyClass {
- // field, constructor, and
- // method declarations
- }

类图

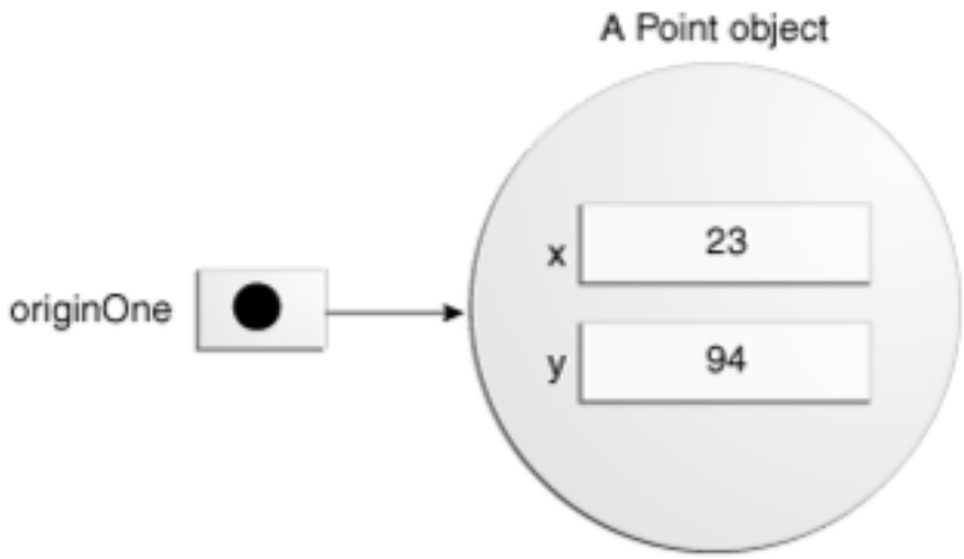


类的定义

```
public class Bicycle {

    private int cadence;
    private int gear;
    private int speed;

    public Bicycle(int startCadence, int startSpeed, int startGear) {  //构造方法
        gear = startGear;
        cadence = startCadence;
        speed = startSpeed;
    }

    public int getCadence() {
        return cadence;
    }

    public void setCadence(int newValue) {
        cadence = newValue;
    }

    。 。 。

    public int getSpeed() {
        return speed;
    }

    public void applyBrake(int decrement) {
        speed -= decrement;
    }

    public void speedUp(int increment) {
        speed += increment;
    }
}
```

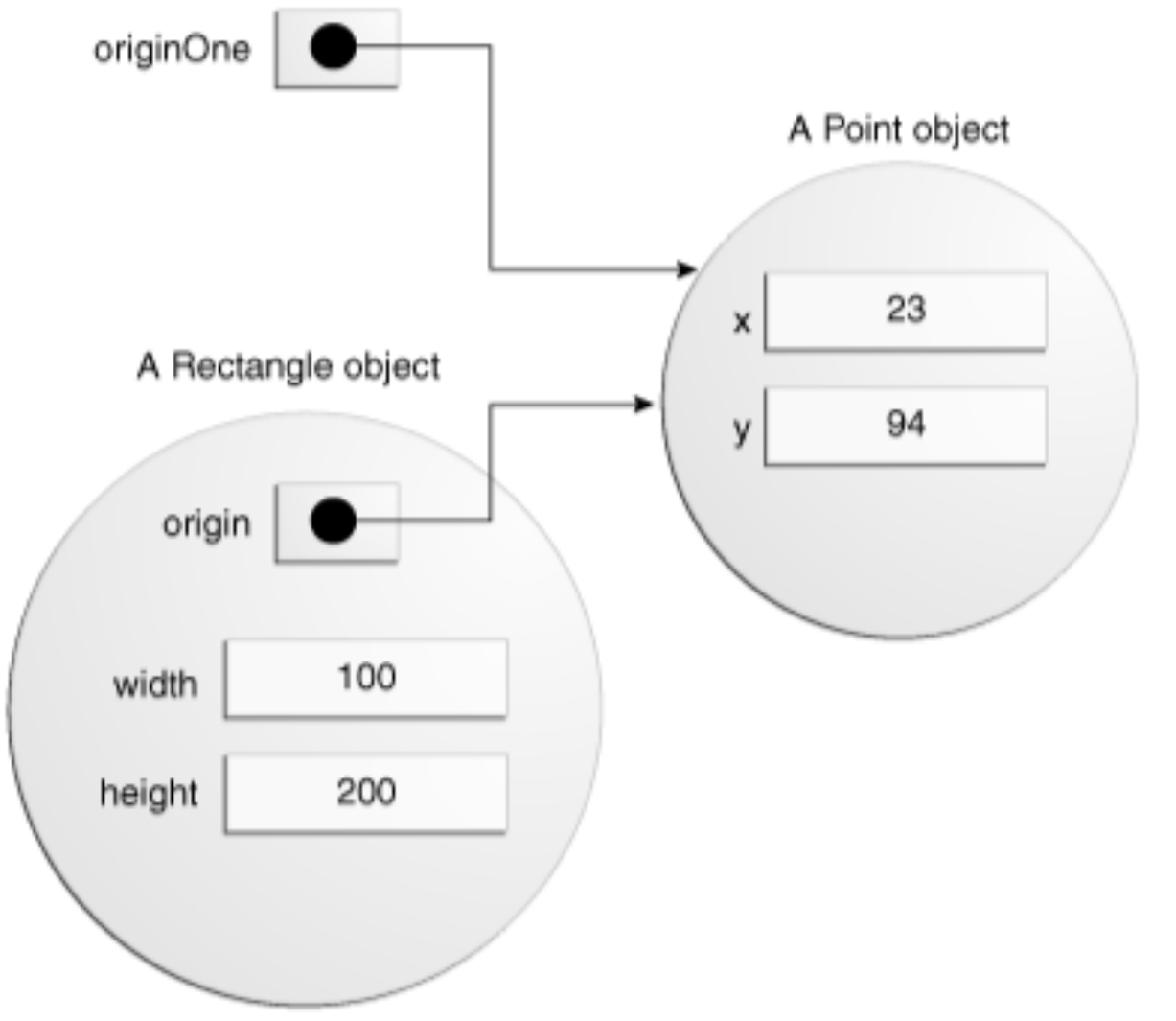
类的定义

```
public class Point {
    public int x = 0;
    public int y = 0;
    //constructor
    public Point(int a, int b) {
        x = a;
        y = b;
    }
}
```




```
Point originOne = new Point(23, 94);
```

```
public class Rectangle {
    public int width = 0;
    public int height = 0;
    public Point origin;

    // four constructors
    public Rectangle() {
        origin = new Point(0, 0);
    }
    public Rectangle(Point p) {
        origin = p;
    }
    public Rectangle(int w, int h) {
        origin = new Point(0, 0);
        width = w;
        height = h;
    }
    public Rectangle(Point p, int w, int h) {
        origin = p;
        width = w;
        height = h;
    }

    // a method for moving the rectangle
    public void move(int x, int y) {
        origin.x = x;
        origin.y = y;
    }

    // a method for computing the area
    // of the rectangle
    public int getArea() {
        return width * height;
    }
}
```



引用变量

变量和方法的访问

- 引用变量.变量名
  - 1 . int height = new Rectangle().height;
  - 2. Rectangle rect = new Rectangle();
  - int height = rect.height;
- 引用变量.方法名（）；
  - System.out.println("Area of rectOne: " + rectOne.getArea());
  - ...
  - rectTwo.move(40, 72);