# 对象

**1、编写一个Conversions对象，加入 inchesToCentimeters,gallonsToLiters 和 milesToKilometers 方法**

**2、定义一个 Point 类和一个伴生对象,使得我们可以不用 new 而直接用 Point(3,4)来构造 Point 实例 apply 方法的使用**

**3、编写一个 Scala 应用程序,使用 App 特质,以反序打印命令行参数,用空格隔开。举例来说,scala Reverse Hello World 应该打印 World Hello**

**4、编写一个扑克牌 4 种花色的枚举,让其 toString 方法分别返回♣,♦,♥,♠，并实现一个函数,检查某张牌的花色是否为红色**

# 继承

**1、扩展如下的BankAccount类，新类CheckingAccount对每次存款和取款都收取1美元的手续费**  
**class BankAccount(initialBalance:Double){  
 private var balance = initialBalance  
 def deposit(amount:Double) = { balance += amount; balance}  
 def withdraw(amount:Double) = {balance -= amount; balance}  
}**

**2、扩展前一个练习的BankAccount类，新类SavingsAccount每个月都有利息产生(earnMonthlyInterest方法被调用)，并且有每月三次免手续费的存款或取款。在earnMonthlyInterest方法中重置交易计数。**

**3、设计一个Point类，其x和y坐标可以通过构造器提供。提供一个子类LabeledPoint，其构造器接受一个标签值和x,y坐标,比如:new LabeledPoint(“Black Thursday”,1929,230.07)**

**4、定义一个抽象类Shape，一个抽象方法centerPoint，以及该抽象类的子类Rectangle和Circle。为子类提供合适的构造器，并重写centerPoint方法**

**5、提供一个Square类，扩展自java.awt.Rectangle并且是三个构造器：一个以给定的端点和宽度构造正方形，一个以(0,0)为端点和给定的宽度构造正方形，一个以(0,0)为端点,0为宽度构造正方形**