

# 模拟扑克牌游戏的洗牌

实验项目命名 javase-exp-02

## 0.1 实验要求

结合面向对象设计原则，分析和设计模拟扑克牌游戏的洗牌过程，完成代码编写。

1. 定义一个表示扑克牌的类 Poke，假设每副牌为 52 张，不考虑包含大小王。其无参构造方法用于默认创建 1 副牌，有参数构造方法用于创建多副牌。定义枚举类 Suit 来表示扑克牌的可选花色，枚举类 Face 来表示扑克牌的可选牌面值。定义集合容器存放所有扑克牌实例，可以使用 ArrayList 容器类，要求适用泛型约束该容器中元素只能为 Card 类型，示例如下：

```
1 ArrayList<Card> cards = new ArrayList<Card>();
```

2. 定义一个表示每张扑克牌的类 Card，用枚举类 Suit 来表示扑克牌的花色，用枚举类 Face 来表示扑克牌的牌面值，实现 Comparable 接口定义牌面的排序方法（暂时不考虑花色和牌面一起的排序规则）。
3. 假设扑克牌游戏需要 2 付（或 2 付以上）的牌，请编程实现模拟洗牌，以及将这些扑克牌分配给参加游戏的所有人，并将参加扑克牌游戏的所有人手中的扑克牌排序后输出。
4. 定义一个 Action 接口，其中规划扑克牌游戏普遍具备的洗牌（shuffle）、发牌（distribute）、显示牌（display）方法，这些方法的参数及返回值类型请根据实验要求的上下文自行设计。
5. 定义一个 PokeGame 类，在此类中包含程序的 main 方法，在该方法中完成游戏洗牌的整个流程；该类需要实现 Action 接口中规划的所有方法。PokeGame 类的代码框架如下：

```
1 package ouc.cs.java.pokegame;  
2  
3 import java.util.Scanner;  
4  
5 public class PokeGame {  
6  
7     public static void main(String[] args) {  
8         System.out.println("该扑克游戏需要几付扑克牌？");  
9         Scanner sc = new Scanner(System.in);
```

```
10     int num = sc.nextInt();
11     // ... 创建牌
12     System.out.println("该扑克游戏有几个玩家？");
13     Scanner sp = new Scanner(System.in);
14     int per = sp.nextInt();
15     // ... 洗牌、发牌
16     System.out.println("\n显示所有的牌：");
17     // ... 显示所有牌
18     System.out.println("\n显示分配给每个人的牌：");
19     // ... 显示分配给每个人的牌
20     System.out.println("\n从所有牌中随机抽出2张牌，比较其大小：");
21     // ... 比较随机抽牌的大小
22 }
23 }
```

6. 要随机产生某个范围内的整数，可以用 `java.util.Random` 类的 `nextInt(int num)`。例如，输出 0 - 51 范围内的一个整数如下：

```
1 Random rand = new Random();
2 int num = rand.nextInt(52);
```

## 0.2 实验过程、步骤及原始记录

实验过程和代码如下：

```
1 .
2 .
3 .
4 .
5 .
6 .
7 .
8 .
9 .
10 .
```

代码运行结果截图如下：  
(请附能够说明代码运行整个流程及结果的几张图片)