# 随手科技笔试题解答

## 题目：

面向对象思想，写一个锄大地扑克牌程序。

一副新的扑克牌，除去大小鬼剩下52张牌（这副牌包括梅花、方块、红桃和黑桃4种花色，每种花色又有13种牌，分别是A、2、3、4、5、6、7、8、9、10、J、Q、K）。发牌者把这52张牌随机发给4个玩家，直到把所有牌发完。按上面要求写一个发牌程序，要求把4个玩家手里的牌都打印出来。

要求：

实现上面程序，并画出UML图。（API可用伪代码）

目的：

1）主要考察面向对象的设计能力。

2）UML知识。

## 解答：

## UML图

该程序对应UML图如下图1所示：

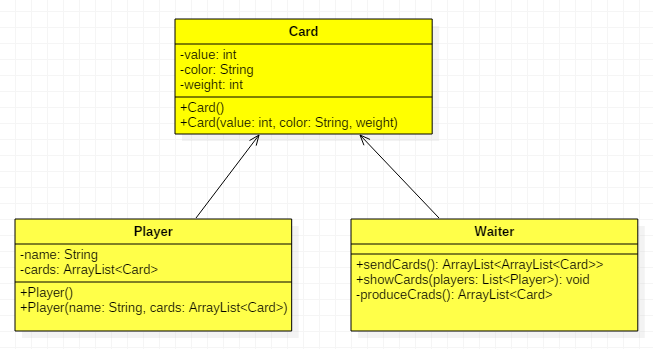


图 1 UML图

## 程序源代码：

该程序源代码托管于GitHub地址：<https://github.com/HarchiCaibao/suiExam>

源代码如下所示：

### 2、1 Card

package com.sui.po;

/\*\*

\* 2018.5.29

\*

\* @author 王汉志

\* 扑克牌对象

\*/

public class Card {

private int value; // 扑克牌的面值

private String color; // 扑克牌的花色

private int weight; // 随机一个数字，用于后期扑克牌的打乱

public int getWeight() {

return weight;

}

public void setWeight(int weight) {

this.weight = weight;

}

public String getColor() {

return color;

}

public int getValue() {

return value;

}

public void setValue(int value) {

this.value = value;

}

public void setColor(String color) {

this.color = color;

}

public Card() {

}

public Card(int value, String color, int weight) {

this.value = value;

this.color = color;

this.weight = weight;

}

}

### 2、2 Player

package com.sui.po;

/\*\*

\* 2018.5.29

\* @author 王汉志

\* 玩家对象

\*/

import java.util.ArrayList;

public class Player {

private String name; // 玩家名称

private ArrayList<Card> cards; // 玩家手里的牌

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public ArrayList<Card> getCards() {

return cards;

}

public void setCards(ArrayList<Card> cards) {

this.cards = cards;

}

public Player() {

}

public Player(String name, ArrayList<Card> cards) {

this.name = name;

this.cards = cards;

}

}

### 2、3 Waiter

package com.sui.po;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

/\*\*

\* 2018.5.29

\*

\* @author 王汉志

\* 发牌服务员

\*/

public class Waiter {

/\*\*

\* 发牌程序 使用最简单的冒泡排序对牌的权值排序实现打乱牌的次序

\*

\* @return 返回四套次序随机的扑克牌

\*/

public ArrayList<ArrayList<Card>> sendCards() {

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

// 得到有花色和面值的52张牌

cards = produceCrads();

// 打乱牌的次序，使用最简单的冒泡排序对权值排序即可实现

boolean flag = true;

int n = cards.size() - 1;

for (int i = 0; i < n && flag; i++) {

flag = false;

for (int j = 0; j < n - i; j++) {

Card c1 = cards.get(j);

Card c2 = cards.get(j + 1);

if (c1.getWeight() > c2.getWeight()) {

// 做标记

flag = true;

// 交换

cards.set(j, c2);

cards.set(j + 1, c1);

}

}

}

// 将乱序的牌平均分成四套

ArrayList<ArrayList<Card>> cardList = new ArrayList<>();

for (int i = 0; i < 4; i++) {

ArrayList<Card> c = new ArrayList<>();

for (int j = i \* 13; j < 13 + i \* 13; j++) {

c.add(cards.get(j));

}

cardList.add(c);

}

return cardList;

}

/\*\*

\* 打印玩家手里的牌

\*

\* @param player 待打印牌的玩家

\*/

public void showCards(List<Player> players) {

if (players.size() == 0) {

System.out.println("参数错误");

return;

}

for (int i = 0; i < players.size(); i++) {

Player player = players.get(i);

System.out.print("玩家" + player.getName() + "手中的牌为： ");

ArrayList<Card> cards = player.getCards();

for (int j = 0; j < cards.size(); j++) {

String color = cards.get(j).getColor();

int v = cards.get(j).getValue();

String value;

if (v == 11) {

value = "J";

} else if (v == 12) {

value = "Q";

} else if (v == 13) {

value = "K";

} else {

value = String.valueOf(v);

}

System.out.print(color + value + " ");

}

System.out.println(" ");

}

}

/\*\*

\* 产生52张牌的程序，牌的顺序一定

\*

\* @return 返回顺序一定的52张牌

\*/

private ArrayList<Card> produceCrads() {

String[] colors = { "黑桃", "红桃", "方块", "梅花" };

int[] values = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13 };

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

Random random = new Random();

for (int i = 0; i < colors.length; i++) {

for (int j = 0; j < values.length; j++) {

// 随机权值

int w = random.nextInt(10000) % (10000 - 1 + 1) + 1;

Card card = new Card(values[j], colors[i], w);

cards.add(card);

}

}

return cards;

}

}

### 2、4 PlayCardTest

package com.sui.test;

import java.util.ArrayList;

import java.util.List;

import com.sui.po.Card;

import com.sui.po.Player;

import com.sui.po.Waiter;

/\*\*

\* 功能测试类 2018.5.29

\*

\* @author 王汉志

\*

\*/

public class PlayCardTest {

public static void main(String[] args) {

// 发牌者

Waiter waiter = new Waiter();

// 玩家

List<Player> players = new ArrayList<>();

// 发牌到每个玩家手中

ArrayList<ArrayList<Card>> cards = waiter.sendCards();

for (int i = 0; i < 4; i++) {

players.add(new Player(String.valueOf(i), cards.get(i)));

}

// 打印每位玩家手中的牌

waiter.showCards(players);

}

}

## 程序运行结果：

图2 程序运行结果