

# 廈門大學



## 信息学院软件工程系

### 《JAVA 程序设计》实验报告

#### 实验 2

姓名：黄勛

学号：22920212204392

学院：信息学院

专业：软件工程系

完成时间：2023.3.7

## 一、实验目的及要求

- 练习 Java 的基本语法
- 熟悉控制结构

## 二、实验题目及实现过程

### (一) 解题过程（基本+拓展）

题目大意：完成博饼游戏流程，请按照游戏规则，编程实现：随机生成六个筛子点数并判断得奖情况，注意需要考虑到多个奖项的情况，比如四进带一秀。



	【一秀】即秀才 有一粒为4点
	【二举】即举人 有两粒为4点
	【四进】即进士 有四粒点数相同（4除外）
	【三红】即探花 有三粒的点数为4点
	【对堂】即榜眼 六粒骰子为123456
	【状元】 有四粒骰子为4点即是状元
	【四红】四个4
	【五子】五个2或3或5或6
	【五王】五个4
	【状元插金花】四个4、二个1
	【六勃红】六个4
	【六勃黑】六个2或3或5或6
	【遍地锦】六个1

拓展部分具体要求如下：

- (1) 提醒用户输入玩家数（6-10）。
- (2) 循环为每个玩家生成六个筛子点数（1-6），根据上图的规则判断所产生的骰子对应的奖项，并输出。
- (3) 游戏结束时（所有奖项已经出完），输出每个玩家所获得的奖项以及每个奖项的个数。

代码编写:

- 生成骰子点数 (保留了测试入口, 可以自行输入)

```
12     private static int[] randomDice() { //随机产生六个骰子点数
13         var dice = new int[7];
14         if (test == false) {
15             Random rnd = new Random();
16             for (int i = 0; i < 6; i++) {
17                 dice[rnd.nextInt( bound: 6) + 1]++;
18             }
19             for (int i = 1; i <= 6; i++)
20                 dice[0] = Math.max(dice[0], dice[i]);
21         } else {
22             dice = new int[7];
23             for(int i=0; i<6; i++)
24             {
25                 int temp = scan.nextInt();
26                 if(temp<1||temp>6)
27                 {
28                     System.out.println("输入错误, 请重新输入");
29                     i--;
30                 }
31                 else
32                     dice[temp]++;
33             }
34             for (int i = 1; i <= 6; i++)
35                 dice[0] = Math.max(dice[0], dice[i]);
36         }
37         return dice;
38     }
```

- 判断是否为状元

```
private static boolean isZhuangYuan(int[] dice) { //判断是否为状元
    return dice[4] >= 4 || dice[0] >= 5;
}
```

- 判断普通奖项

```
private static int[] judgePrize(int[] dice) { //判断普通奖项
    if (isZhuangYuan(dice)) {
        if (dice[4] == 1) //五子登科带一秀
        {
            int[] ret = {1, 6};
            return ret;
        } else {
            int[] ret = {1};
            return ret;
        }
    } else {
        if (dice[0] == 1) //对堂
        {
            int[] ret = {2};
            return ret;
        } else if (dice[0] == 4) //四进
        {
            if (dice[4] > 0) {
                int[] ret = {4, 7 - dice[4]};
                return ret;
            } else {
                int[] ret = {4};
                return ret;
            }
        } else if (dice[4] > 0) //三红二举一秀
        {
            int[] ret = {7 - dice[4]};
        }
    }
}
```

……代码过长截取了部分，具体在实验附带 Lab2/src/中查看

- 判断状元奖项类型

```
private static int[] distinguishZhuangYuan(int[] dice) { //判断状元奖项类型
    int[] ret = new int[3];
    if (dice[1] == 2) //状元插金花
        ret[0] = 1;
    else if (dice[4] == 6) //六博红
        ret[0] = 2;
    else if (dice[0] == 6) //六博黑
    {
        ret[0] = 3;
        for (int i = 1; i <= 6; ++i) {
            if (dice[i] == 6) {
                ret[1] = i;
                break;
            }
        }
    }
    else if (dice[4] == 5) //五王
    {
        ret[0] = 4;
        for (int i = 1; i <= 6; ++i) {
            if (dice[i] != 0 && i != 4) {
                ret[1] = i;
                break;
            }
        }
    }
    else if (dice[0] == 5) //五子登科
    {
        ret[0] = 5;
        for (int i = 1; i <= 6; ++i) {
            if (dice[i] != 0 && i != 4) {
                ret[1] = i;
                break;
            }
        }
    }
}
```

- 最后确认唯一的状元得主

```
130     private static int[] ChampionTime; //记录玩家投到状元的轮次
      1个用法
131     private static int getBestPlayer() { //最后确认唯一的状元得主
132         int ret = 0, id = -1;
133         int[] best = {7, 0, 0};
134         for (int[] i : Champion) {
135             ++id;
136             if (i == null) continue;
137             int result[] = distinguishZhuangYuan(i);
138             if (result[0] > best[0]) continue;
139             else if (result[0] == best[0]) {
140                 if (result[1] < best[1]) continue;
141                 else if (result[1] == best[1]) {
142                     if (result[2] < best[2]) continue;
143                     else if (result[2] == best[2]) {
144                         if (ChampionTime[id] > ChampionTime[ret]) continue;
145                     }
146                 }
147             }
148             ret = id;
149             best = result;
150         }
151         return ret;
152     }
```

- 判断游戏是否结束（还有没有剩余奖项）

```
private static boolean isGameOver() { //判断游戏是否结束（还有没有剩余奖项）
    int sum = 0;
    for (int i : Remain_Prize)
        sum += i;
    if (sum == 0)
        return true;
    else
        return false;
}
```

- Main 方法编写游戏全流程

```
164 ▶ public static void main(String[] args) {
165
166     //变量初始化
167     int round = 1, player = 0, Total_Player;
168     //是否手动测试
169     System.out.println("是否手动测试? 1:是 0:否");
170     int select = scan.nextInt();
171     if(select == 1)
172     {
173         test = true;
174     }
175     //输入玩家数
176     System.out.printf("请输入玩家数(6-10): ");
177     Total_Player = scan.nextInt();
178     while (true) {
179         if (Total_Player > 10 || Total_Player < 6) {
180             System.out.println("玩家数量范围限制为6-10, 请重新输入:");
181         } else {
182             break;
183         }
184         Total_Player = scan.nextInt();
185     }
186
187     System.out.printf("当前为第 %d 轮, 生成第 %d 位玩家的点数: ", round, player);
188     int[] Dice = randomDice();
189     for (int i = 1; i <= 6; i++) {
190         for (int j = 1; j <= Dice[i]; j++)
191             System.out.printf("%d ", i);
192     }
193     System.out.print('\n');
194     int[] prizeids = judgePrize(Dice);
195     System.out.print("骰子对应的奖项: ");
196     for (int prizeid : prizeids) {
197         if (prizeid == 1) {
198             var temp = distinguishZhuangYuan(Dice);
199             System.out.printf("%s ", TopPrizeStr[temp[0]]);
200             Champion[player] = Dice;
201             Remain_Prize[1] = 0;
202             ChampionTime[player] = round * (Total_Player + 1) + player;
203         } else {
204             System.out.format("%s ", BasePrizeStr[prizeid]);
205             if (prizeid > 0) {
206                 if (Remain_Prize[prizeid] > 0) {
207                     --Remain_Prize[prizeid];
208                     ++PrizeRecord[player][prizeid];
209                 } else
210                     System.out.format("( %s已全部领完) ", BasePrizeStr[prizeid]);
211             }
212         }
213     }
214 }
```

```

229 //寻找状元
230 int finalChampion = getBestPlayer();
231 System.out.format("最终的状态是第 %d 位玩家, 在第 %d 轮时投出了%s,点数为: "
232 , finalChampion, ChampionTime[finalChampion] / (Total_Player + 1),
233 TopPrizeStr[distinguishZhuangYuan(Champion[finalChampion])[0]]);
234 for (int i = 1; i <= 6; ++i)
235     for (int j = 1; j <= Champion[finalChampion][i]; ++j)
236         System.out.format("%d ", i);
237 System.out.print('\n');
238 ++PrizeRecord[finalChampion][1];
239
240 //输出所有玩家获得的奖项
241 for (int i = 1; i <= Total_Player; ++i) {
242     System.out.format("第 %d 位玩家获得:", i);
243     for (int j = 1; j <= 6; ++j) {
244         if (PrizeRecord[i][j] > 0) {
245             System.out.format("%d个%s ", PrizeRecord[i][j], BasePrizeStr[j]);
246         }
247     }
248     System.out.print('\n');
249 }

```

## (二) 结果展示

- 打开测试入口：

```

"D:\Program Files\Java\jdk-17\bin\java.exe" "-javaa
是否手动测试? 1:是 0:否
1
请输入玩家数(6-10): 6
当前为第 1 轮, 生成第 1 位玩家的点数: 4 4 4 4 4 5
4 4 4 4 4 5
骰子对应的奖项: 五王
当前为第 1 轮, 生成第 2 位玩家的点数: 1 2 3 4 5 6
1 2 3 4 5 6
骰子对应的奖项: 对堂
当前为第 1 轮, 生成第 3 位玩家的点数: 4 4 4 4 1 2
1 2 4 4 4 4
骰子对应的奖项: 状元

```

- 关闭测试入口并随机生成骰子：

```

运行: MoonCakeGame x
"D:\Program Files\Java\jdk-17\bin\java.exe" "-javaagent:D
是否手动测试? 1:是 0:否
0
请输入玩家数(6-10): 8
当前为第 1 轮,生成第 1 位玩家的点数: 1 3 4 4 5 6
骰子对应的奖项: 二举
当前为第 1 轮,生成第 2 位玩家的点数: 1 3 4 4 5 6
骰子对应的奖项: 二举
当前为第 1 轮,生成第 3 位玩家的点数: 1 2 2 3 3 5
骰子对应的奖项: 无
当前为第 1 轮,生成第 4 位玩家的点数: 4 4 4 5 5 5
骰子对应的奖项: 三红
当前为第 1 轮,生成第 5 位玩家的点数: 1 1 1 2 2 6
骰子对应的奖项: 无
当前为第 1 轮,生成第 6 位玩家的点数: 1 3 4 5 5 5
骰子对应的奖项: 一秀
当前为第 1 轮,生成第 7 位玩家的点数: 1 1 3 6 6 6
    
```

```

运行: MoonCakeGame x
当前为第 20 轮,生成第 7 位玩家的点数: 2 2 3 3 3 3
骰子对应的奖项: 无
当前为第 26 轮,生成第 8 位玩家的点数: 2 3 3 4 6 6
骰子对应的奖项: 一秀 (一秀已全部领完)
当前为第 27 轮,生成第 1 位玩家的点数: 3 4 5 5 5 5
骰子对应的奖项: 四进 一秀 (一秀已全部领完)
最终的状态是第 1 位玩家,在第 12 轮时投出了五子登科,点数为: 4 6 6 6 6 6
第 1 位玩家获得:1个状元 1个三红 1个四进 4个二举 2个一秀
第 2 位玩家获得:1个三红 2个四进 2个二举 4个一秀
第 3 位玩家获得:1个对堂 2个四进 1个二举 3个一秀
第 4 位玩家获得:1个三红 1个二举 4个一秀
第 5 位玩家获得:2个二举 3个一秀
第 6 位玩家获得:1个对堂 1个四进 3个二举 6个一秀
第 7 位玩家获得:1个四进 1个二举 6个一秀
第 8 位玩家获得:1个三红 1个四进 2个二举 4个一秀

进程已结束,退出代码0
    
```



### 三、实验总结与心得记录

通过本次实验，在博饼模拟过程中，我们需要使用 Java 语言来实现整个游戏的逻辑，这需要对 Java 基本语法和控制结构有一定的了解。

Java 基本语法包括变量声明、数据类型、运算符、条件语句、循环语句等等。在博饼模拟过程中，我们需要定义多个变量来表示骰子点数、博饼结果、玩家得分等信息。同时，我们需要使用运算符进行加减乘除等运算，以及使用条件语句和循环语句来控制游戏的逻辑。

控制结构是编程中非常重要的一部分，包括顺序结构、分支结构和循环结构。在博饼模拟过程中，我们需要使用分支结构和循环结构来实现游戏的逻辑。例如，在进行博饼时，如果骰子点数为“五子”，则直接计分并结束本轮游戏；如果骰子点数为“状元插金花”，则计分并再次掷骰子。我们需要使用 if 语句来实现这些逻辑。

同时，我们也需要使用循环结构来实现游戏的多轮进行。例如，当玩家选择开始游戏后，我们需要不断进行骰子掷点和计分的过程，直到玩家选择结束游戏为止。在这里，我们需要使用 while 语句来实现。

总之，博饼模拟过程是一个较为简单的 Java 编程练习，通过实现游戏的逻辑，可以加深对 Java 基本语法和控制结构的理解和掌握。