

Java 实现 3D 福彩模拟实验报告

软件学院软件工程专业 2019 级 2 班 韦诗睿 1913184

一、实验题目

(一) 主要内容：使用 Java 运用类和对象内容实现给定的福彩玩法。

(二) 主要功能：

1. 实现不同的福彩玩法。用户选择玩法，输入投注内容，可以返回奖金数量
2. 判断输入是否合法。不同的玩法有不同的输入规则，需要验证输入是否符合该玩法的规则。

二、设计思路

(一) 流程分析：

用户选择给出的福彩玩法之一并输入投注内容，根据不同福彩玩法的投注方式判断用户的投注内容是否正确；若不正确则返回重新输入，若正确则根据不同玩法的规则，判断输入内容和由系统产生的三个随机数表示的中奖号码之间的关系是否正确；若正确则输出相应的奖金数目，若不正确则输出奖金数目为 0。

由上分析，总的流程对于每种福彩玩法大致相同，主要包括：

1. 系统生成获奖号码：设计 `getwinNumber` 函数进行随机数生成，`winNumber` 数组记录中奖数字。
2. 获取用户的输入并判断用户输入是否合法：利用 `getinput` 函数获取用户的输入，`userInput` 和 `userNumber` 记录用户输入并将输入为数字的字符串转化为 `int` 类型数组，`isAvailable` 函数对输入内容的合法性进行判断。
3. 判定是否中奖并输出奖金：利用 `judwin` 函数依据规则对是否获奖进行判断，`getWins` 函数获取奖金金额并输出。

综上，不同投注方式在流程上的相似性，使得设计类 `Lottery3D` 来定义所需的基本成员变量和函数 `play` 来完成总流程在操作上可行。

(二) 功能分析：

给定的福彩种类有：`Single`、`Group`、`OneD`、`Guess1D`、`TwoD`、`Guess2D`、`General`、`Sum`、`Package`、`LargeSmall`（猜大小）、`ThreeSame`（猜三同）、`Tractor`

（拖拉机）、OddEven（猜奇偶）。不同的福彩玩法虽有大致相同的流程，但之间仍有较大区别，分别表现在：

1. 不同福彩的输入内容：single、group、general、package 玩法以三个长度的数字作为投注内容；Guess1D、Guess2D、sum 玩法以长度为一或二的数字作为投注内容；OneD、TwoD 玩法输入包含数字和星号(*)的三个字符；LargeSmall、ThreeSame、Tractor、OddEven 玩法以英文如“Large”“Small”、“Yes”“No”为输入内容进行选择。因此需要根据不同的玩法对输入的合法性进行判断。

2. 不同的判定方法：具体内容可参考《福彩 3D 游戏设奖方案详解》。本人在对游戏判定分析中，将判断内容具体分为数字、位置和个数三类。数字即为对中奖和投注号码的数字是否相同进行判定；位置即为对中奖和投注号码的三位数字的排列位置是否符合规定进行判定；个数即对中奖和投注号码中数字是否重复出现和重复出现的个数是否相同进行判定。如 single 玩法要求号码完全相同、位置相同，则为数字、位置和个数三个参数的综合判定，因此难度最高，奖金也最高；OneD 玩法是对某一位置的号码进行投注，即为对数字和位置两个参数的判定；group 玩法要求号码相同而位置随意，则是对数字和个数两个参数进行判定。其他类的分析也同样包含且仅包含这三个参数。因此可以设置不同的数组对不同的信息进行记录，userNumber、winNumber 数组记录数字，countwin、countinput 数组记录个数，设置 numbercount 函数对数字的个数进行处理，location 数组记录位置信息。

3. 不同的奖金内容：不同的福彩玩法由于中奖概率的不同而返回不同的奖金数量。

（三）类设计分析：

1. 整体流程和不同玩法种类之间类的设计：

（1）由上分析可知，由于流程的一致性可减少代码的冗余，可以设计 Lottery3D 作为基类定义所需的成员变量和所需的流程函数；不同福彩玩法则定义不同的类，在继承 Lottery3D 类的同时对其其中的一些成员函数如判断输入合法性的 isAvailable 函数、判断是否中奖的 judwin 函数、返回奖金的 getWins 函数进行重写。选择不同的投注玩法即生成相应玩法的类的对象，并在相同的 play 函数中调用子类重写的成员函数。

如：Guess1D 类和 Tractor 类对判断合法性的 isAvailable 函数的重写：

```
1. // Guess1D.isAvailable()
2. //只输入一个数字
3. boolean isAvailable(String s) {
4.     //长度检查
5.     if(s.length()!=getstancount()) {return false;}
6.     //判断输入是否为数字
7.     for(int i = 0;i<getstancount();i++) {
8.         char a = s.charAt(i);
9.         if(a<48||a>57) {
10.            return false;
11.        }
12.    }
13.    return true;
14. }
```

```
1. //Tractor.isAvailable()
2. //判断文字输入是否合法
3. boolean isAvailable(String s) {
4.     //设置给定的文字输入范围
5.     choice[0] = "Yes";
6.     choice[1] = "No";
7.     //对文字的输入是否在规定范围内进行检查
8.     return this.CharaLegal(s);
9. }
```

（2）保持流程一致性：

在对函数进行重写时，既需要根据不同的规则对函数进行具体实现，又需要考虑到 play 函数整体流程的一致性。如在 play 函数中，对输入判定是否合法后需要将其赋值给已定义的记录数字的成员数组 userNumber 中：此时若长度只有一个或两个数字，或输入带有星号（*）、文字等特殊字符，则无法被转换为数字存入数组中。为了保持该流程的一致性，在对合法性判断完成后，且对后续获奖判定不加影响的情况下对输入的字符进行处理：如将星号（*）转换为数字-1 存入 Integer 类型的 userNumber 数组；将长度不足 3 的数字输入加上相应数量的星号（*）变为 3 个长度的输入再进行上一步的转化；文字判定后将文字输入转化为“000”存入数组，采取另一个 label 变量以 0、1 值的形式对输入类型的正负进行记录，保证转换 String 的一致性。

如将 String 类型转化为 Integer 类型数组时对星号（*）的设置：

```

1. //Lottery3D.StringToIntArray(String s)
2. //将 String 类型的转化为 Integer 类型数组
3. Integer[]StringToIntArray(String s){
4.     Integer[]arr = new Integer[length];
5.     for(int i = 0;i<length;i++) {
6.         if(s.charAt(i)=='*') {
7.             //星号设置为-1
8.             arr[i] = -1;
9.         }else{
10.            arr[i] = Integer.parseInt(s.substring(i,i+1));
11.        }
12.    }
13.    return arr;
14. }

```

如对输入字符串采取保持长度的处理：

```

1. //Lottery3D.play()
2. //保持长度的一致性
3. if(Input.length()==1) {
4.     Input+="*";
5. }else {
6.     if(Input.length()==2) {
7.         Input+="*";
8.     }

```

2. 不同玩法种类之间的关系设计：

(1) 根据《福彩 3D 游戏设奖方案详解》，部分玩法种类输入的合法性判定和获奖判定其实具有相似性。如 OneD 和 TwoD 玩法中，皆为允许其中的 Stancount 个位置有数字（stancount<3），其他位置为星号（*），只需要在不同类中设定 stancount 的值传入 isAvaiable 函数中即可对合法性进行相同的判定，因此 TowD 可继承自 OneD。再如 Single 和 General 通选 1 皆为对三个数字一一对应判定，因此 General 可继承 judwin 函数自 Single 再另外实现 General 通选 2。

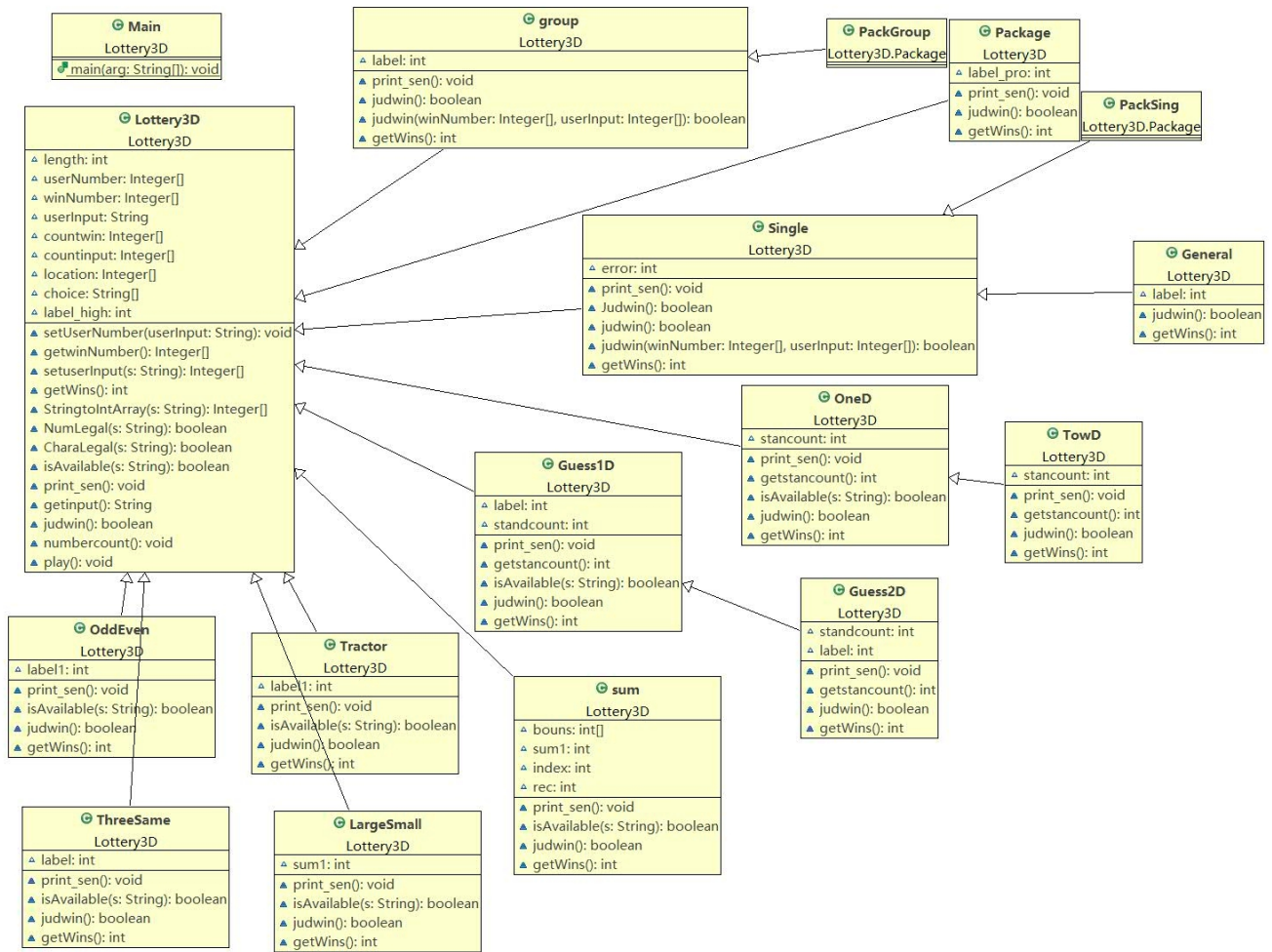
不同玩法种类相互继承的有：General extends Single，Guess2D extends Guess1D，TwoD extends OneD。

(2) 内部类实现多重继承。

在（1）分析的基础上，由于包选 Package 玩法包括全中和组中，因此判定时既包含 Single 玩法的判定也包含 Group 玩法的判定。为了实现别的类的代码复用，在 Package 类的内部分别定义 PackSing 类和 PackGroup 类继承 Single 和 Group，以在 Package 中调用两个类的判定函数。

```
1.  //Package.judwin()
2.  boolean flag1 = true;
3.  //先判断是否全中
4.  PackSing ps = new PackSing();
5.  flag1 = ps.judwin(this.winNumber,this.userNumber);
6.  if(flag1) {
7.      if(count==2) {
8.          label_pro = 1;
9.      }else {
10.         label_pro = 3;
11.     }
12. }else {
13.     //再判断是否组中
14.     PackGroup pg = new PackGroup();
15.     flag1 = pg.judwin(this.winNumber,this.userNumber);
16.     if(flag1) {
17.         if(count==2) {
18.             label_pro = 2;
19.         }else {
20.             label_pro = 4;
21.         }
22.     }
23. }
24. return flag1;
```

3. 类图（由 Eclipse 的 AmaterasUML 生成）



（四）实现的功能及其他设计：

1. Main 程序的主要交互：

- （1）选择投注方式时将不同种类投注方式的名称列出，方便用户输入。
- （2）本程序实现了《福彩 3D 游戏设奖方案详解》的所有福彩玩法，增加了用户选择。
- （3）可以由用户选择是否再来一次，输入 “Yes” 或 “No” 进行选择。
- （4）当投注方式不存在时则提示重新输入。

```
请输入投注方式
普通投注方式有：
SingleChoose,
OneD(TwoD)Choose,
GroupChoose,
Guess1D(Guess2D)Choose,
GeneralChoose,PackageChoose
高级投注方式有：
SumChoose,
LargeSmallChoose,
ThreeSameChoose,
TractorChoose,
OddEvenChoose
sing
您输入的投注方式不存在，请重新输入
SingleChoose
中奖号码为：1 3 7
请输入000~999之间的整数
137
您获得的奖金为1040
再来一次？
Yes
请输入投注方式
普通投注方式有：
SingleChoose,
OneD(TwoD)Choose,
GroupChoose,
Guess1D(Guess2D)Choose,
GeneralChoose,PackageChoose
高级投注方式有：
SumChoose,
LargeSmallChoose,
ThreeSameChoose,
TractorChoose,
OddEvenChoose
```

2. Lottery3D 的 play 基本流程：

- （1）根据不同福彩游戏规则输出不同的输入规则提示，如 Single 玩法中提示为“请输入 000~999 之间的整数”，Tractor 玩法中输入 “Yes” “No”。
- （2）为了测试程序设计测试模块，即先输出生成的中奖号码后接收用户输入。

(3) 根据不同的福彩游戏规则判断输入是否合法，包括数字合法性判断（数字长度、星号判断）和文字合法性判断（是否在可以选择的文字范围内），若不合法则提示重新输入。

(4) 判定是否中奖并输出奖金。

LargeSmallChoose

中奖号码为：2 3 5

请输入“Large”或“Small”

sma

请输入正确的投注内容

请输入“Large”或“Small”

Small

您获得的奖金为6

再来一次？

OneDChoose

中奖号码为：5 3 4

请输入确定位置的一个数字，其他位输入*，例如，如果确定个位数为2，请输入**2

5**

您获得的奖金为10

再来一次？

OddEvenChoose

中奖号码为：8 1 8

请输入“Odd”或者“Even”123

请输入正确的投注内容

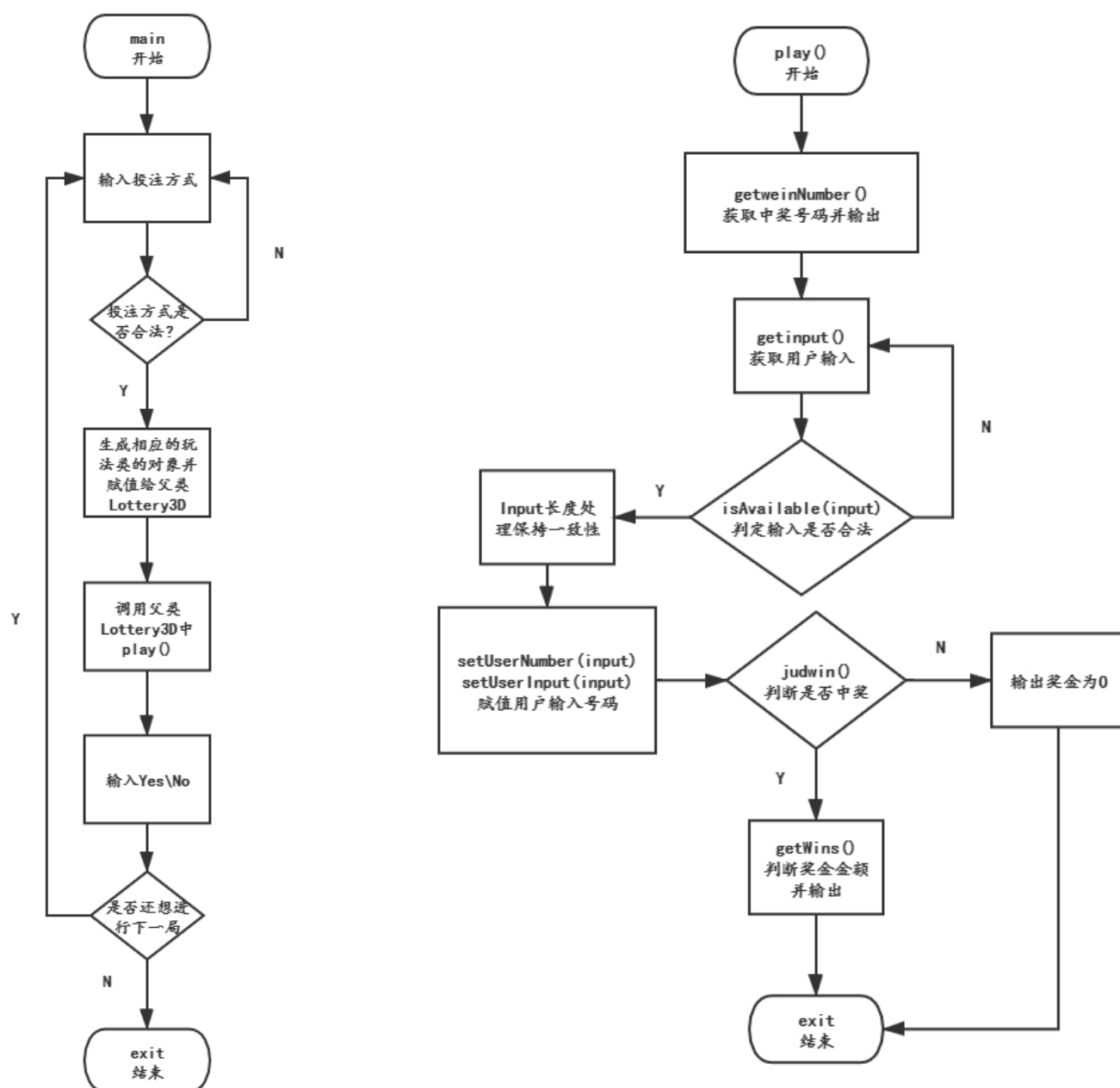
请输入“Odd”或者“Even”Odd

您获得的奖金为0

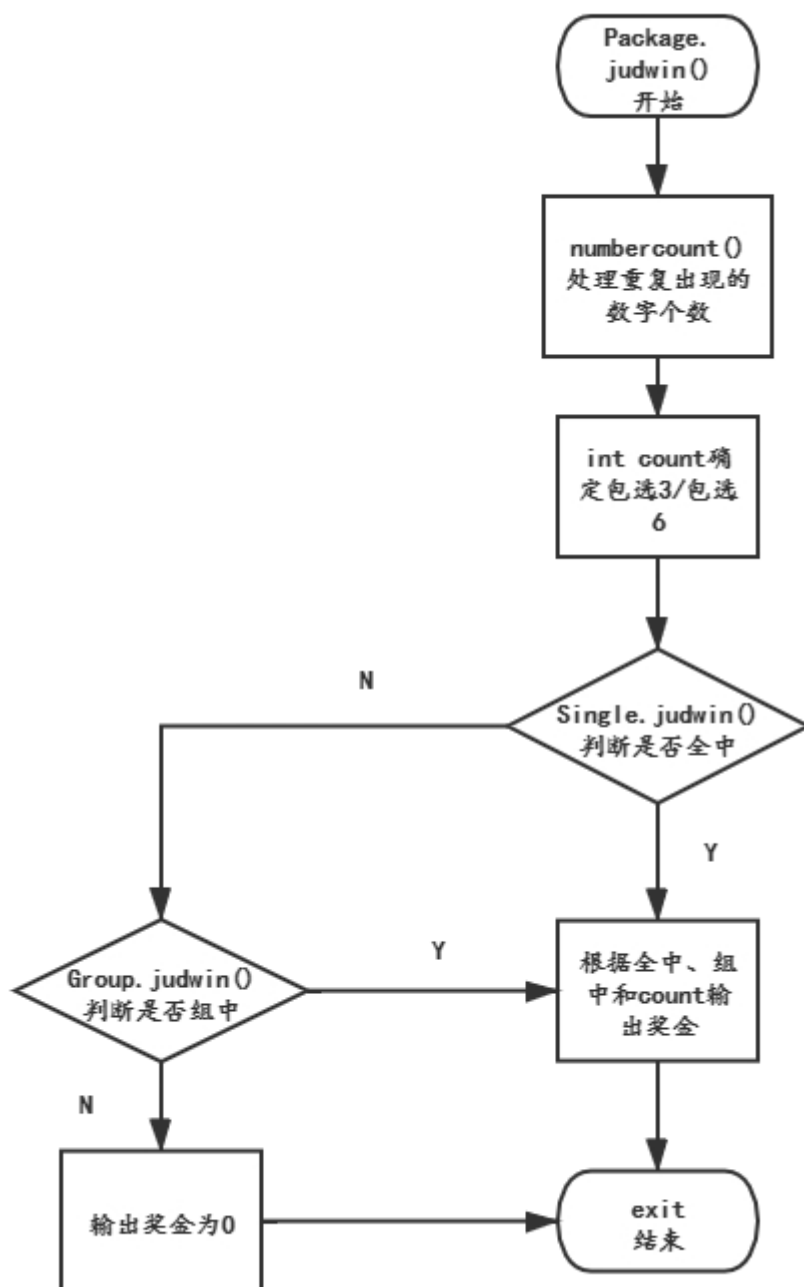
再来一次？

三、关键代码流程图

1. Main()以及 Lottery3D.play()



2. Package.judwin(): 包含内部类的调用。



四、测试案例设计

（一）测试投注方式是否符合输入：

投注方式输入规范为：

SingleChoose,OneD(TwoD)Choose,GroupChoose,Guess1D(Guess2D)Choose,
GeneralChoose,PackageChoose, SumChoose,LargeSmallChoose,ThreeSameChoose,
TractorChoose,OddEvenChoose

1. 和文字无关的字符：123、\$%^#、***、ruiruirui

```
123
您输入的投注方式不存在，请重新输入
$%^#
您输入的投注方式不存在，请重新输入
***
您输入的投注方式不存在，请重新输入
ruiruirui
您输入的投注方式不存在，请重新输入
```

2. 大小写敏感测试：只有符合规定的大写方式才判断合法：single、sum、singlechoose

```
single
您输入的投注方式不存在，请重新输入
sum
您输入的投注方式不存在，请重新输入
singlechoose
您输入的投注方式不存在，请重新输入
```

3. 正确的输入方式测试：SingleChoose、TractorChoose

```
SingleChoose
|中奖号码为：6 7 1
|请输入000~999之间的整数
```

```
TractorChoose
|中奖号码为：9 1 5
|请输入“yes”或者“no”|
```

（二）测试不同福彩玩法的输入是否正确

测试案例设计：（“√”即为合法性判断后改玩法允许输入的字符串值）

长度测试：1、12、123

其他字符设计：\$%^ 、rui

带星号输入设计：***、12*、1**

文字设计：Yes、Large、Odd

案例	Single	One	Two	Group	G1D	G2D	General	Package	Sum	LS	Three	Tractor	OddE
1					√				√				
12						√			√				
123	√			√			√	√					
\$%^													
ru													

12*			√										
1**		√											
Yes											√	√	
Large										√			
Odd													√

1. SingleChoose、GroupChoose、GeneralChoose、PackageChoose：三种玩法皆为输入三个数字

测试样例：

(1) 长度测试：12 1 123

(2) 数字测试：ru \$%^ 12* ***

2. OneD、TwoD 输入带星号的长度为 3 的字符串：

测试样例：

(1) 长度测试&其他特殊字符测试：12 %\$^ ru

(2) 星号测试：1** 12* ***

3. Guess1D、Guess2D 和 Sum：不带星号的长度为 1 或 2 的字符串（sum<=27）：

测试样例：1 12

4. LargeSmall、Tractor、OddEven、ThreeSame: 输入文字进行选择

测试样例: Large、Yes、Odd、No

(三) 测试不同福彩玩法的判定和奖金输出是否正确: 选择其中的五种玩法, 依照随机生成的中奖号码进行测试。

Single:

```
SingleChoose
中奖号码为: 0 2 5
请输入000~999之间的整数
025
您获得的奖金为1040
```

OneD:

```
OneDChoose
中奖号码为: 6 1 7
请输入确定位置的一个数字, 其他位输入*, 例如, 如果确定个位数为2, 请输入**2
6**
您获得的奖金为10
```

General:

```
GeneralChoose
中奖号码为: 1 3 4
请输入000~999之间的整数
143
您获得的奖金为0
```

Sum:

```
SumChoose
中奖号码为: 6 9 9
请输入三个数之和
24
您获得的奖金为104
```

OddEven:

```
OddEvenChoose
中奖号码为: 0 7 3
请输入“Odd”或者“Even”Odd
您获得的奖金为0
```

五、心得与收获

(一) 知识:

1. 类之间的设计: 在初次得到详细的福彩方案玩法时, 首先是惊叹于玩法种类之多、判断之繁琐, 但后来经过分析后得出了数字、个数和位置三种信息, 便以这三个共性参数出发设计主要的 play 函数和 judwin 函数流程。在按顺序编写代码的过程中, 逐步发现一些福彩玩法的判断存在共性, 可以直接继承已编写完成的类而无需每次继承自基类 Lottery3D。

2. 为了保持函数流程的一致性，添加了很多对输出参数进行长度处理、字符处理的其他函数。虽然主函数的流程以及代码得到了简化，但仍对这一处理方法带来的潜在的不确定的 bug 存在担忧，希望今后遇到同类项目时可以再认真分析不同输入之间的关联，找到解决代码冗余的另一种可能性。

3. debug 方面曾遇到的一些问题：

（1）成员函数有多态性，但成员变量没有多态性。当为了实现代码复用而通过设置其中的某些参数达到在子类中调用父类函数的目的时，不能直接用 `this.变量名` 调用子类的成员变量，因为这时调用的是父类的成员函数，其中的变量也是父类的成员变量。因此需通过复写另一成员函数 `Get()` 获取子类成员变量进行调用。

（2）内部类：在外部类定义内部类的对象时，虽然内部类并不是直接继承基类，但是在调用构造函数时通过内部类的父类，仍会调用基类的构造函数，造成外部类的一些已经赋值的成员变量的值被 `null` 或 `0` 重新覆盖。此时需要再调用内部类的时候传入外部类的已经赋值的成员变量，避免值的丢失。

（3）形参和实参：向函数中传递参数时是值传递，因此修改的只是形参而不是实参。`String` 类型同样也是，`StringBuffer` 可以解决这一问题。

（4）使用 `String` 类型时，注意 `==` 是对同一性的判定，判定是否为同一对象；而大部分情况下使用的是 `String.equals` 方法。

（二）其他：

在 Debug 时通过和电脑进行交互慢慢地将整个游戏的过程完善，而不是枯燥地解决语法问题，是和以前写算法题不一样的 debug 感受。通过自己的设计完成不同种类福彩规则的实现，很有成就感。

在分析的过程中还尝试使用正在学习的概率论知识对福彩中奖概率进行分析，如 `single` 玩法中奖概率为 $1/1000$ ，`group` 玩法组选 3 时中奖概率为 $3/1000$ ，组选 6 时为 $6/1000$ ，`OneD` 为 $1/30$ ，`guess1D` 猜中 1 为 $1/10$ 等等。在后续的过程中将会考虑实现通过计算不同选择的期望值，基于自己设定的一注彩票的投注金额，对投注方式进行机器推荐。

六、源代码

(一) main 函数

```
1. package Lottery3D;
2. import java.util.*;
3.
4. //实现不同的福彩玩法
5. //不同的玩法有不同的输入规则，需要验证输入是否合法
6. public class Main {
7.     public static void main(String arg[]) {
8.
9.         outer:
10.        while(true){
11.            //选择投注方式
12.            //可以查看投注方式
13.            System.out.print("请输入投注方式"+"\\n");
14.            System.out.print("普通投注方式有：\\n");
15.            System.out.print("SingleChoose,\\nOneD(TwoD)Choose,\\nGroupChoose,\\nGuess1D(Gue
ss2D)Choose,\\nGeneralChoose,PackageChoose\\n");
16.            System.out.print("高级投注方式有：\\n");
17.            System.out.print("SumChoose,\\nLargeSmallChoose,\\nThreeSameChoose,\\nTractorChoo
se,\\nOddEvenChoose\\n");
18.            Scanner sc = new Scanner(System.in);
19.            String type = "";
20.
21.            Lottery3D l = new Lottery3D();
22.            //设置循环避免投注方式不合法
23.            mylabels:
24.            while(true) {
25.                type = sc.next();
26.                switch(type) {
27.                    case "SingleChoose":l = new Single();break mylabels;
28.                    case "OneDChoose":l = new OneD();break mylabels ;
29.                    case "GroupChoose":l = new group();break mylabels;
30.                    case "Guess1DChoose":l = new Guess1D();break mylabels;
31.                    case "TwoDChoose":l = new TowD();break mylabels;
32.                    case "Guess2DChoose":l = new Guess2D();break mylabels;
33.                    case "GeneralChoose":l = new General();break mylabels;
34.                    case "SumChoose":l = new sum();break mylabels;
35.                    case "PackageChoose":l = new Package();break mylabels;
36.                    case "LargeSmallChoose":l = new LargeSmall();break mylabels;
37.                    case "ThreeSameChoose":l = new ThreeSame();break mylabels;
38.                    case "TractorChoose":l = new Tractor();break mylabels;
39.                    case "OddEvenChoose":l = new OddEven();break mylabels;
```

```

40. default: System.out.print("您输入的投注方式不存在, 请重新输入\n");
41. }
42. }
43. l.play();
44.
45. inner:
46. while(true) {
47. System.out.print("再来一次? "+"n");
48. Scanner sc1 = new Scanner(System.in);
49. String type1 = sc1.next();
50. if(type1.equals("Yes")) {
51. break inner;
52. }else {
53. if(type1.equals("No")) {
54. break outer;
55. }else {
56. continue inner;
57. }
58. }
59. }}}}

```

(二) 基类 Lottery3D:

```

1. package Lottery3D;
2. import java.util.*;
3.
4. //3D 福彩基本定义
5. class Lottery3D {
6. //成员变量定义
7. int length = 3;//基本的长度定义
8. Integer[]userNumber = new Integer[length];
9. Integer[]winNumber = new Integer[length];
10. String userInput;
11. Integer[]countwin = new Integer[11];//对号码中数字出现的个数进行计数
12. Integer[]countinput = new Integer[11];
13. Integer[]location = new Integer[3];//对号码的位置进行记录
14. String []choice = new String[2];//对需要输入文字的投注种类的文字进行记录
15. int label_high = 0;//标志文字类的投注, 0 表示否定、偶数等
16. //成员函数定义
17.
18. //将输入传入
19. void setUserNumber(String userInput){
20. this.userInput = userInput;
21. }
22. //随机设置中奖号码
23. Integer[]getwinNumber(){

```



```

24. Integer[] arr = new Integer[length];
25. Random r = new Random();
26. for(int i = 0; i < length; i++) {
27.     arr[i] = r.nextInt(10);
28. }
29. return arr;
30. }
31.
32. //赋值用户输入 号码
33. Integer[] setUserInput(String s) {
34.     //转化字符串
35.     char a = ' ';
36.     for(int i = 0; i < s.length(); i++) {
37.         a = s.charAt(i);
38.         //字母时, 为了保持 play 函数的统一性, 变成数字
39.         if((a < 48 || a > 57) && a != '*') {
40.             s = "000";
41.             break;
42.         }
43.     }
44.     return this.StringToIntArray(s);
45. }
46.
47. //判断奖金
48. //需要被不同的玩法重写
49. int getWins() {
50.     return 0;
51. }
52.
53. //将 String 类型的转化为 Integer 类型数组
54. Integer[] StringToIntArray(String s) {
55.     Integer[] arr = new Integer[length];
56.     for(int i = 0; i < length; i++) {
57.         if(s.charAt(i) == '*') {
58.             //星号设置为-1
59.             arr[i] = -1;
60.         } else {
61.             arr[i] = Integer.parseInt(s.substring(i, i+1));
62.         }
63.     }
64.     return arr;
65. }
66. //数字的非法性检查
67. boolean NumLegal(String s) {

```

```

68.     int length_pro = s.length();
69.     boolean flag = true;
70.     char a = ' ';
71.     for(int i = 0;i<length_pro;i++) {
72.         a = s.charAt(i);
73.         if(a<48||a>57) {
74.             flag = false;
75.             break;
76.         }
77.     }
78.     return flag;
79. }
80. //字符的非法性检查
81. boolean CharaLegal(String s) {
82.     if(!s.equals(choice[0])&&!s.equals(choice[1])) {
83.         return false;
84.     }
85.     if(s.equals(choice[0])) {
86.         label_high = 1;
87.     }
88.     return true;
89. }
90.
91. //判断输入是否合法
92. //需要被不同的玩法的 isAvailable 函数重写
93. //默认输入：三个数字，只设计对数字地非法性检查
94. boolean isAvailable(String s) {
95.     //长度检查
96.     if(s.length()!=length) {return false;}
97.     //对数字的非法性进行检查
98.     return this.NumLegal(s);
99. }
100. //不同输入有着不同的输入语句
101. //不同类进行重写
102. void print_sen() {
103.
104. }
105. //获取用户投注
106. String getinput() {
107.     print_sen();
108.     Scanner sc1 = new Scanner(System.in);
109.     String s = sc1.next();
110.     return s;
111.

```

```

112. }
113. //判断是否获胜
114. //不同投注方式有不同的规则
115. boolean judwin() {
116.     return true;
117. }
118. //对个数进行处理
119. void numbercount(){
120.     //初始化 count 数组为0
121.     for(int i = 0;i<11;i++) {
122.         this.countinput[i] = this.countwin[i] = 0;
123.     }
124.     //记录每个数字出现的次数
125.     for(int i = 0;i<length;i++) {
126.         if(userNumber[i]==-1) {
127.             this.countinput[10]++;
128.         } else {
129.             this.countinput[this.userNumber[i]]++;
130.         }
131.         this.countwin[this.winNumber[i]]++;
132.     }
133. }
134.
135. //投注进程：包括测试模块
136. void play() {
137.     //获取中奖号码
138.     this.winNumber = this.getwinNumber();
139.     System.out.print("中奖号码
        为: "+String.valueOf(this.winNumber[0])+" "+String.valueOf(this.winNumber[1])+" "+String.
        ng.valueOf(this.winNumber[2])+"\n");
140.     //初始值定义
141.     String Input = "";
142.     //判断输入是否合法
143.     while(true) {
144.         //输入提示
145.         Input = this.getinput();
146.         if(this.isAvailable(Input)) {
147.             //保持长度的一致性
148.             if(Input.length()==1) {
149.                 Input+="*";
150.             } else {
151.                 if(Input.length()==2) {
152.                     Input+="*";
153.                 }

```

```

154.     }
155.     break;
156. }else {
157.     System.out.print("请输入正确的投注内容\n");
158. }
159. }
160.
161. //将输入传入
162. this.setUserNumber(Input);
163. //赋值输入 号码
164. this.userNumber = this.setuserInput(Input);
165. //判断是否中了
166. if(judwin()) {
167.     //判断奖金
168.     int bouns = this.getWins();
169.     //输出奖金
170.     System.out.print("您获得的奖金为"+String.valueOf(bouns)+"\n");
171. }else {
172.     //没获奖， 输出奖金为 0
173.     System.out.print("您获得的奖金为"+String.valueOf(0)+"\n");
174. }
175. }
176. }

```

(三) 不同类:

1. Single&General:

```

1.     package Lottery3D;
2.
3.     //单选
4.     //对三个号码以唯一的排列方式进行投注
5.     //购买号码与开奖号码完全相同， 且位置相同
6.     class Single extends Lottery3D{
7.         int error = 0;//记录错误数
8.         //不同输入有着不同的输入语句
9.         void print_sen() {
10.             System.out.print("请输入 000~999 之间的整数\n");
11.         }
12.
13.         //判断是否获奖
14.         //既保持了 Lottery3D 的一致性， 也可以使通选拓展
15.         boolean Judwin() {
16.             boolean flag = true;
17.             for(int i = 0;i<length;i++) {
18.                 if(this.winNumber[i]!=this.userNumber[i]) {

```

```

19.     flag = false;
20.     error++;
21. }
22. }
23. return flag;
24. }
25. boolean judwin() {
26.     return Judwin();
27. }
28. boolean judwin(Integer[]winNumber,Integer[]userInput) {
29.     this.winNumber = winNumber;
30.     this.userNumber = userInput;
31.     return Judwin();
32. }
33. //获取奖金:
34. int getWins() {
35.     return 1040;
36. }
37.
38. }
39. //通选:
40. //两次中奖机会, 先中通选 1, 若不中可再中通选 2
41. class General extends Single{
42.     int label = 0;//记录是通选 1 还是通选 2, 默认什么都不是, 1-为通选 1, 2 为通选 2
43.
44.     boolean judwin() {
45.         boolean flag_pro = true;
46.         flag_pro = Judwin();
47.         if(flag_pro) {
48.             label = 1;
49.         }else {
50.             if(error==1) {
51.                 label = 2;
52.             }
53.         }
54.         if(label==0) {
55.             flag_pro = false;
56.         }else {
57.             flag_pro = true;
58.         }
59.         return flag_pro;
60.     }
61.     //获取奖金
62.     int getWins() {

```

```

63.     if(label == 1) {
64.         return 470;
65.     } else {
66.         return 21;
67.     }
68. }
69. }

```

2. Group:

```

1.     package Lottery3D;
2.
3.     //组选
4.     class group extends Lottery3D {
5.
6.         //组选 3 和组选 6 的标志
7.         int label = 1; //0-为组选 3 (有两个数字相同), 1-为组选 6 (三个数字都不同)
8.         void print_sen() {
9.             System.out.print("请输入 0~999 之间的整数\n");
10.        }
11.        //判断是否中奖
12.        boolean judwin() {
13.            //需要个数时进行处理
14.            numbercount();
15.            boolean flag = true;
16.            for(int i = 0; i < length; i++) {
17.                //若存在个数不相同, 则为 False
18.                if(!(countwin[this.winNumber[i]] == countinput[this.winNumber[i]])) {
19.                    flag = false;
20.                    break;
21.                } else {
22.                    if(countwin[this.winNumber[i]] >= 2) {
23.                        label = 0;
24.                    }
25.                }
26.            }
27.            return flag;
28.        }
29.        boolean judwin(Integer[] winNumber, Integer[] userInput) {
30.            this.winNumber = winNumber;
31.            this.userNumber = userInput;
32.            return judwin();
33.        }
34.        //奖金
35.        int getWins() {
36.            if(label == 1) {

```

```

37.     return 173;
38. }else {
39.     return 346;
40. }
41. }
42. }

```

3. OneD&TwoD:

```

1.  package Lottery3D;
2.  //1D: 对百位、十位或个位中某一特定位置上的号码进行投注
3.  class OneD extends Lottery3D {
4.      int stancount = 1;
5.      //不同输入有着不同的输入语句
6.      void print_sen() {
7.          System.out.print("请输入确定位置的一个数字，其他位输入*，例如，如果确定个位
            数为 2，请输入**2\n");
8.      }
9.      int getstancount() {
10.         return stancount;
11.     }
12.     //判断输入是否合法
13.     //1D 只允许有一个位置存在数字，其他位置为星号
14.     boolean isAvailable(String s) {
15.         //长度检查
16.         if(s.length()!=length) {return false;}
17.         boolean flag = true;
18.         int count = 0;
19.         char a = ' ';
20.         int j = 0;
21.         for(int i = 0;i<length;i++) {
22.             a = s.charAt(i);
23.             //是数字
24.             if(a>=48&&a<=57) {
25.                 count++;
26.                 location[j] = i;
27.                 j++;
28.             }else {
29.                 //若输入了别的符号，也算非法
30.                 if(a!=42) {
31.                     flag = false;
32.                     return flag;
33.                 }
34.             }
35.         }
36.         if(count!=getstancount()) {

```

```

37.     flag = false ;
38. }else {
39.     flag = true;
40. }
41.     return flag;
42. }
43.
44. //判断是否获奖
45. boolean judwin() {
46.     boolean flag = true;
47.     if(this.winNumber[location[0]]!=this.userNumber[location[0]]) {
48.         flag = false;
49.     }
50.     return flag;
51. }
52. //获取奖金
53. int getWins() {
54.     return 10;
55. }
56. }
57.
58. //2D:买 2 个号码，数字正确，位置相同
59. class TowD extends OneD{
60.     int stancount = 2;
61.     //不同输入
62.     void print_sen() {
63.         System.out.print("请输入确定位置的两个数字，其他位输入*，例如，如果确定十位
           数和个位数分别为 1 和 2，请输入*12\n");
64.     }
65.     //获取成员变量
66.     int getstancount() {
67.         return stancount;
68.     }
69.     //判断是否获奖
70.     boolean judwin() {
71.         boolean flag = true;
72.         for(int i = 0;i<this.stancount;i++) {
73.             if(this.winNumber[location[i]]!=this.userNumber[location[i]]) {
74.                 flag = false;
75.             }
76.         }
77.         return flag;
78.     }
79.     //获取奖金

```



```

80. int getWins() {
81.     return 104;
82. }
83. }

```

4. Guess1D&Guess2D:

```

1. package Lottery3D;
2. import java.util.Scanner;
3.
4. class Guess1D extends Lottery3D {
5.     //记录是猜中1 或2 或3
6.     int label = 0; //默认没猜中
7.     int standcount = 1;
8.
9.     void print_sen() {
10.        System.out.print("请输入一个数字");
11.    }
12.    int getstandcount() {
13.        return standcount;
14.    }
15.
16.    //只输入一个数字
17.    boolean isAvailable(String s) {
18.        //长度检查
19.        if(s.length() != getstandcount()) {return false;}
20.        //判断输入是否为数字
21.        for(int i = 0; i < getstandcount(); i++) {
22.            char a = s.charAt(i);
23.            if(a < 48 || a > 57) {
24.                return false;
25.            }
26.        }
27.        return true;
28.    }
29.    //判断输赢并确定猜中几
30.    boolean judwin() {
31.        boolean flag = true;
32.        //位置处理
33.        numbercount();
34.        label = this.countwin[this.userNumber[0]];
35.        if(label == 0) {
36.            flag = false;
37.        }
38.        return flag;
39.    }

```

```

40. //奖金数量
41. int getWins() {
42.     switch(label) {
43.         case 1:return 2;
44.         case 2:return 12;
45.         default:return 230;
46.     }
47. }
48. }
49.
50. class Guess2D extends Guess1D{
51.     int standcount = 2;
52.     int label = 1 ;//判断是两同号还是两不同号： 1-默认为两不同号， 2-为两同号
53.     //输入
54.     void print_sen() {
55.         System.out.print("请输入两个数字");
56.     }
57.     int getstandcount() {
58.         return standcount;
59.     }
60.     //判断输赢并确定猜中几
61.     boolean judwin() {
62.         boolean flag = true;
63.         //个数处理
64.         numbercount();
65.         for(int i = 0;i<this.standcount;i++) {
66.             if(this.countinput[this.userNumber[i]]==this.countwin[this.userNumber[i]]) {
67.                 if(this.countinput[this.userNumber[i]]>=2) {
68.                     label = 2;
69.                 }
70.             }else {
71.                 flag = false;
72.             }
73.         }
74.         return flag;
75.     }
76.     //获取奖金
77.     int getWins() {
78.         if(label==1) {
79.             return 19;
80.         }else {
81.             return 37;
82.         }
83.     }

```

84. }

5. Package:

```
1. package Lottery3D;
2. //包选
3. //内部类进行多态性继承
4. public class Package extends Lottery3D {
5.     int label_pro = 0; //记录是哪种获奖方式，默认什么都不是
6.     //包选3 全中: 1
7.     //包选3 组中: 2
8.     //包选6 全中: 3
9.     //包选6 组中: 4
10.
11. void print_sen() {
12.     System.out.print("请输入 0~999 之间的整数\n");
13. }
14. //一个类继承 Single 类
15. //三位全部相同
16. class PackSing extends Single {
17. }
18. //一个类继承 group 类
19. //可以不顾顺序
20. class PackGroup extends group {
21. }
22. //判断是否获奖
23. boolean judwin() {
24.     int count = 1; //每个数都不相同
25.     //处理个数
26.     this.numbercount();
27.     //确定个数
28.     for(int i = 0; i < length; i++) {
29.         if(this.countwin[this.winNumber[i]] >= 2) {
30.             count = 2; //存在两个数相同
31.         }
32.     }
33.     boolean flag1 = true;
34.     //先判断是否全中
35.     PackSing ps = new PackSing();
36.     flag1 = ps.judwin(this.winNumber, this.userNumber);
37.     if(flag1) {
38.         if(count == 2) {
39.             label_pro = 1;
40.         } else {
41.             label_pro = 3;
42.         }
```

```

43.     }else {
44.         //再判断是否组中
45.         PackGroup pg = new PackGroup();
46.         flag1 = pg.judwin(this.winNumber,this.userNumber);
47.         if(flag1) {
48.             if(count==2) {
49.                 label_pro = 2;
50.             }else {
51.                 label_pro = 4;
52.             }
53.         }
54.     }
55.     return flag1;
56. }
57. //获取奖金
58. int getWins() {
59.     int bouns = 0;
60.     switch(label_pro) {
61.         case 1:bouns = 693;break;
62.         case 2:bouns = 173;break;
63.         case 3:bouns = 606;break;
64.         case 4:bouns = 86;break;
65.     }
66.     return bouns;
67. }
68. }

```

6. Tractor&Sum&ThreeSame&OddEven&LargeSmall:

```

1.     package Lottery3D;
2.         //拖拉机
3.         //是指对全部以升序或降序连续排列的号码进行投注（890、098、901、109 除外）
4.         class Tractor extends Lottery3D{
5.             int label1 = 0;//记录猜的开和不开
6.
7.             //输出
8.             void print_sen() {
9.                 System.out.print("请输入“Yes”或者“No”");
10.            }
11.            //判断文字输入是否合法
12.            boolean isAvailable(String s) {
13.                //设置给定的文字输入范围
14.                choice[0] = "Yes";
15.                choice[1] = "No";
16.                //对文字的输入是否在规定范围内进行检查
17.                return this.CharaLegal(s);

```

```

18. }
19. //判断是否获奖
20. //拖拉机的规则为升序或者降序都可以。
21. boolean judwin() {
22.     //处理个数
23.     this.numbercount();
24.     int a = this.winNumber[0];
25.     int b = this.winNumber[1];
26.     int c = this.winNumber[2];
27.     //判断升序
28.     if(a==b-1&&a==c-2&&b==c-1) {
29.         label1 = 1;
30.     }
31.     //判断降序
32.     if(c==b-1&&c==a-2&&a-1==b) {
33.         label1 = 1;
34.     }
35.     if(label_high == label1) {
36.         return true;
37.     }else {
38.         return false;
39.     }
40. }
41.
42. //获得奖金
43. int getWins() {
44.     return 65;
45. }
46.
47. }
48.
49. //猜奇数还是偶数
50. class OddEven extends Lottery3D{
51.     int label1 = 0;//默认为偶数
52.     //输出
53.     void print_sen() {
54.         System.out.print("请输入“Odd”或者“Even”");
55.     }
56.     //判断输入是否合法
57.     boolean isAvailable(String s) {
58.         choice[0] = "Odd";
59.         choice[1] = "Even";
60.         return this.CharaLegal(s);
61.     }

```

```

62. //判断是否获奖
63. boolean judwin() {
64. //处理个数
65. this.numbercount();
66. int i = 1;
67. int j = 0;
68. int count1 = 0;//计算奇数
69. int count2 = 0;//计算偶数
70.
71. while(j<this.countwin.length-2&& i<this.countwin.length-1) {
72. count1+=this.countwin[i];
73. count2+=this.countwin[j];
74. i+=2;
75. j+=2;
76. }
77. //奇数
78. if(count1==3&&label_high==1) {
79. return true;
80. }else {
81. //偶数
82. if(count2==3&&label_high==0) {
83. return false;
84. }
85. }
86. return false;
87. }
88. //取得奖金
89. int getWins() {
90. return 8;
91. }
92. }
1. //和数
2. class sum extends Lottery3D {
3. //不同和数的奖金数量
4. int []bouns = {1040,345,172,104,69,49,37,29,23,19,16,15,15,14};
5. int sum1 = 0;
6. int index = 0;
7. int rec = 0;
8.
9. //不同输入有着不同的输入语句
10. void print_sen() {
11. System.out.print("请输入三个数之和\n");
12. }
13. //输入检查

```

```

14. boolean isAvailable(String s) {
15.     boolean flag = true;
16.     //数字的非法性检查
17.     flag = this.NumLegal(s);
18.
19.     if(flag) {
20.         //转化数值
21.         rec = Integer.parseInt(s);
22.         //数值不在 0~27 之间
23.         if(rec<0||rec>27) {
24.             flag = false;
25.         }
26.     }
27.     return flag;
28. }
29. //判断是否中奖
30. boolean judwin() {
31.     boolean flag = true;
32.     //计算获奖的数
33.     for(int i = 0;i<length;i++) {
34.         sum1 = sum1+this.winNumber[i];
35.     }
36.     if(rec!=sum1) {
37.         flag = false;
38.     }
39.     return flag;
40. }
41. //获取奖金
42. int getWins() {
43.     if(sum1>13) {
44.         index = 27-sum1;
45.     }else {
46.         index = sum1;
47.     }
48.     return bouns[index];
49. }
50.
51. }
52.
53. //猜大小
54. class LargeSmall extends Lottery3D{
55.     int sum1 = 0;
56.     //不同输入有着不同的输入语句
57.     void print_sen() {

```

```

58.    System.out.print("请输入“Large”或“Small”\n");
59.    }
60.    //判断输入是否合法
61.    boolean isAvailable(String s) {
62.        choice[0] = "Large";
63.        choice[1] = "Small";
64.        return this.CharaLegal(s);
65.    }
66.    //判断是否中奖
67.    boolean judwin() {
68.        boolean flag = false;
69.        //计算获奖的数
70.        for(int i = 0;i<length;i++) {
71.            sum1 = sum1+this.winNumber[i];
72.        }
73.        if(sum1>=19&&label_high==1) {
74.            return true;
75.        }else {
76.            if(sum1<19&&label_high==0) {
77.                return true;
78.            }
79.        }
80.        return flag;
81.
82.    }
83.    //取得奖金
84.    int getWins() {
85.        return 6;
86.    }
87.    }
1.    //猜三同
2.    class ThreeSame extends Lottery3D{
3.        int label = 0;//记录是否开出豹子数，默认为0-不开，1-开
4.        //输出
5.        void print_sen() {
6.            System.out.print("请输入“Yes”或者“No”");
7.        }
8.        //判断输入是否正确
9.        boolean isAvailable(String s) {
10.            choice[0] = "Yes";
11.            choice[1] = "No";
12.            return CharaLegal(s);
13.
14.        }

```



```
15. //判断是否获奖
16. boolean judwin() {
17. //处理个数
18. this.numbercount();
19. //开出豹子数了
20. if(this.countwin[this.winNumber[0]]==3) {
21. if(label_high==1) {
22. return true;
23. }
24. }else {
25. //没开出豹子数
26. if(label_high==0) {
27. return true;
28. }
29. }
30. return false;
31. }
32. //取得奖金
33. int getWins() {
34. return 104;
35. }
36. }
```

//796 行