

# 浙江大学实验报告

课程名称： Java 应用技术 实验类型： 无

实验项目名称： 火柴棒游戏

学生姓名： 杨樾人 专业： 软件工程 学号： 3160104875

同组学生姓名： 无 指导老师： 鲁伟明

实验地点：          实验日期： 2018 年 10 月 9 日

## 一、实验目的和要求

火柴棒游戏：

1. 用户从命令行输入最大数字的位数（如 1 位数、2 位数、3 位数）；
2. 用户从命令行输入提示数（2 或 3），表示等号左式数字的个数；
3. 用户从命令行输入题目类型编号（移动、移除、添加），以及火柴棒根数；
4. 系统随机自动生成火柴棒游戏，并展示（直接用数字展示）；
5. 用户输入答案，系统验证是否正确；若正确，则提示正确；若错误，则让用户继续输入；
6. 若用户直接回车，则显示正确答案。

## 二、实验内容和原理

1. 使用 Java 进行第一个项目编程；
2. 随机生成符合要求的题目；
3. 对用户输入的答案进行验证



2 最大二位数)  
2 (左边两个数)  
1 (表示移动) 1 (表示1根)

### 三、主要仪器设备

Mac OS 10.13.6 , Eclipse

### 四、操作方法与实验步骤

#### 4.1 类的实现

为实现作业中的要求，共实现了三个 Java 类，分别为 `GameTest`，`Bar` 和 `Present` 类。

`GameTest` 类主要用于与用户的交互，读入用户选择的题目模式，比如最大位数、操作数个数、移动增加或减少、根数等条件。

`Bar` 类为此项目的核心代码，包含符合用户输入条件的题目的生成，检测用户输入答案的正确与否和显示正确答案。

`Present` 类较为简单，是为将数字符号以火柴棒的形式展现出来的一个类。

#### 4.2 `GameTest` 类详解：

`GameTest` 类中包含了 `main` 函数，为程序的入口函数。

此类需先调用 `Bar` 类，`new` 一个 `Bar` 类变量，用户输入最大位数、操作数个数、模式、改变火柴棒的根数，`GameTest` 读入这些条件后传入 `Bar` 类的 `void setCharacter(int maxDigit, int numOfOperators, Character userSymbol, int bar)` 方法，将条件传入到 `Bar` 类之中。

`GameTest` 类随后调用 `Bar` 类中的 `generate` 方法，将生成的题目显示出来。

之后用户输入答案，答案经过 `stringFilter(String str)` 方法过滤到无关字符后，调用 `isEqual()` 方法，检测用户输入的式子是否为等式。如果为等式，将输入的答案再传入 `Bar` 类的 `compare` 方法，与最初生成的题目进行比较，检测是否符合条件。

#### 4.3 Present 类详解：

`Present` 类中存储了各个数字火柴棒的字符串表示形式，每个数字和符合由一个 `7*5` 的 `string` 二维数组表示，`Present` 类的 `presentBars(String input)` 读入字符串后，将字符串用火柴棒的形式展现出来，结果如下图所示。



#### 4.4 Bar 类详解：

`Bar` 类最为复杂，`Bar` 类中有如下方法：

```
public void setCharacter(int maxDigit, int numOfOperators, Character userSymbol, int bar)
public void userInput()
public String showResult()
public boolean isEqual()
public boolean compare()
public void calculateEachBar(String user_input)
public int numberOf1(int n)
public String generate()
public void increaseBarOfOneNum(int totalBar)
public void decreaseBarOfOneNum(int totalBar)
public void tranBarOfOneNum(int totalBar)
public void increaseBarForTrans(int totalBar)
public void decreaseBarForTrans(int totalBar)
public boolean compareForTrans(Character tranSymbol)
```

以下为 `Bar` 中的变量：

```
public String question = "";
public char[] charQuestion = new char[16];
public String answer = "";
public Character symbol = '3'; //choose mode
public int bars = 0; // bars
public int numOfOperators = 2;
public int maxDigit = 3;
public String[] a = new String[] {"***", "***", "***", "***", "*", "*"};
public String[] b = new String[] {"***", "***", "***", "***", "*", "*"};
```

```

public int [] detResult = new int[15]; //difference compare : a and b
public String formalInput = "";
public static Map<Character, Integer> map = new HashMap<Character, Integer>() {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    {
        put('0', 0b1111011);
        put('1', 0b1001000);
        put('2', 0b0111101);
        put('3', 0b1101101);
        put('4', 0b1001110);
        put('5', 0b1100111);
        put('6', 0b1110111);
        put('7', 0b1001001);
        put('8', 0b1111111);
        put('9', 0b1101111);
        put('+', 0b0110000000);
        put('-', 0b0010000000);
        put('=', 0b1010000000);
        put('*', 0);
    }
};

```

现在开始介绍我生成问题的方法。

### 1) 生成等式:

由于最多有 3 个 3 位数, 故一般形式为 $***+***+***=***$ , 所以我用一个 15 位的 char[]来存储这个等式。

在生成等式的时候, 会有最大位数和操作数个数的限制。我有一个 randomGenerate(int maxDigit)方法, 用来生成最大位数之内的数字, 每次是生成三个操作数和两个操作符, 最后检验用户输入的操作数个数, 如果为 2, 就把 char[]数组里的 7-10 位重新置为'', 最后计算出左边的结果, 填到最后三位。至此, 生成等式工作完成。

### 2) 生成题目

题目是在等式基础之上生成的, 我建立了两个二维数组, 一个是每个数字和符号的七段数码管表示方法, 另一个是数字之间转换需要变动几根火柴棒。

```

public static Map<Character, Integer> map = new HashMap<Character, Integer>() {
    /**
     *
     */
    private static final long serialVersionUID = 1L;

    {

```

```

        put('0', 0b1111011);
        put('1', 0b1001000);
        put('2', 0b0111101);
        put('3', 0b1101101);
        put('4', 0b1001110);
        put('5', 0b1100111);
        put('6', 0b1110111);
        put('7', 0b1001001);
        put('8', 0b1111111);
        put('9', 0b1101111);
        put('+', 0b0110000000);
        put('-', 0b0010000000);
        put('=', 0b1010000000);
        put('*', 0);
    }
};

int [][] difference = {
    {0, -4, 0, 0, 0, 0, 0, 0, -3, 1, 0, 0, 0, 0, 0},
    {4, 0, 0, 3, 2, 0, 0, 0, 1, 5, 4, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0},
    {0, -3, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 0, 0},
    {0, -2, 0, 0, 0, 0, 0, 0, 0, 3, 2, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 1, 0, 2, 1, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, -1, 0, 0, 1, 0, 0, 0, 0, 0, 0},
    {3, -1, 0, 2, 0, 0, 0, 0, 0, 4, 3, 0, 0, 0, 0},
    {-1, -5, -2, -2, -3, -2, -1, -4, 0, -1, 0, 0, 0, 0, 0},
    {0, -4, 0, -1, -2, -1, 0, -3, 1, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}
};

```

以下逐个讨论增加、减少、移动火柴棒的实现方法。

增加：

增加火柴棒是在原等式的基础之上逐个数字增加，选择哪一个数字和增加几根火柴棒(**totalBar**)，我有一个随机算法，对于 **char[15]** 的这个存储着等式的数组，每次随机出来一个 1-14 的数字，作为要改变的那个数字的下标。然后查阅二维数组，选取小于等于 **totalBar** 的最大数字，将该位的数字进行改变，**totalBar** 减去增加的火柴棒个数，直到 **totalBar** 为零。为了避免算法陷入死循环，我又创建了一个 **change[15]** 的 **int** 数组，存储每次改变的数字的位置，当这个位置被选中三次以后，将不能再改变。如果该等式无法改变 **totalBar** 个数字，再超出一定随机改变次数后，程序将跳出，重新生成一个等式，避免了死循环。该算法时间复杂度低，对于处理增加 10 根火柴棒也能够轻松跑出结果。

减少：

减少火柴棒是在原等式的基础之上逐个数字减去，选择哪一个数字和增加几根火柴棒(**totalBar**)，有一个随机算法，和增加类似。对于 **char[15]**的这个存储着等式的数组，每次随机出来一个 1-14 的数字，作为要改变的那个数字的下标。然后查阅二维数组，选取小于等于 **totalBar** 的最大负数绝对值，将该位的数字进行改变，**totalBar** 减去增加的火柴棒个数，直到 **totalBar** 为零。为了避免算法陷入死循环，我又创建了一个 **change[15]**的 int 数组，存储每次改变的数字的位置，当这个位置被选中三次以后，将不能再改变。如果该等式无法改变 **totalBar** 个数字，再超出一定随机改变次数后，程序将跳出，重新生成一个等式，避免了死循环。该算法时间复杂度低，对于处理增加 10 根火柴棒也能够轻松跑出结果。

移动：

移动是在增加和减少的基础之上做到的。首先对原等式进行增加 **totalBar** 根火柴棒的操作，之后又对式子进行减少 **totalBar** 根火柴棒的操作，得出的式子即为最后的题目。为了避免对同一个位置的同一根火柴棒既增加又减少，我们可以对新增位置的火柴棒做标记，如果是对此位置的火柴棒进行减少，我们可以让他跳过，重新选择其他位置的火柴棒进行减少。

### 3) 用户输入规范化

由于用户输入的等式，并不是 15 个字符，也没有添加 '\*' 符号，程序在读入用户输入的等式之后，会对等式进行规范化，将其转化为 15 个字符的标准形式，方便后续进行比较。

### 4) 测试答案

首先对用户的答案进行验证等号两侧是否相等。

其次按照异或的规则，对用户输入的规范后的字符串与原等式字符串进行按位异或，可以得到有多少位发生改变，改变的总数即为火柴棒改变的个数。

```
public boolean compare() {
    boolean result = false;

    //          if(formalInput.equals(answer)) {
    //              return true;
    //          }

    int total = 0;
    calculateEachBar(formalInput);
    switch (symbol) {
        //+ bars
```

```

        case '1':
            for(int i = 0; i < 14; i++) {
                if (detResult[i] < 0) {
                    return false;
                }
                else {
                    total = total + detResult[i];
                }
            }
            if (total == bars) {
                result = true;
            }
            break;

        //- bars
        case '0':
            for(int i = 0; i < 14; i++) {
                if (detResult[i] > 0) {
                    return false;
                }
                else {
                    total = total + detResult[i];
                }
            }
            if (total == bars) {
                result = true;
            }
            break;

        // transport bars
        case '2':
            for(int i = 0; i < 14; i++) {
                total = total + detResult[i];
            }
            if (total == 2 * bars) {
                result = true;
            }
            break;
        default:
            break;
    }

    return result;
}

public void calculateEachBar(String user_input) {
    int []temp = new int[16];
    for(int i = 0; i < 15; i++) {
        //          System.out.println(question.charAt(i));
        //          System.out.println(question);
        //          System.out.println(map.get(question.charAt(i)));
        //          System.out.println(map.get(user_input.charAt(i)));
        temp[i] = map.get(question.charAt(i)) ^ map.get(user_input.charAt(i));
        detResult[i] = numberOf1(temp[i]);
    }
}

```

```
//calculate the number of 1
public int numberOf1(int n) {
    int sum = 0;
    while(n != 0) {
        sum++;
        n = n & (n - 1);
    }
    return sum;
}
```

## 五、实验结果与分析

### 5.1 对最大位数进行控制

```
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
2
2
1
2
The question is below: 52-13=43
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
62-19=43
user input the answer:
62-19=43
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
```

figure 1

由最终结果可以看出，该实验顺利完成。

### 5.2 对操作数进行控制

```
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
1
3
1
2
The question is below: 3+8+2=7
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
3+6-2=7
user input the answer:
3+6-2=7
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
```

figure 2 一位



```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
2
2
1
2
The question is below: 52-13=43
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
62-19=43
user input the answer:
62-19=43
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :

```

figure3 两位

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
3
3
0
10
The question is below: 525-101+153=111
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
525-307+753=971
user input the answer:
525-307+753=971
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :

```

figure 4 三位

由最终结果可以看出，该实验顺利完成。

### 5.3 对增加火柴棒进行测试

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
1
3
0
1
The question is below: 3+3-1=11
* * *
* * *
* * *
* * *
* * *
One possible answer of the question is below:
3+9-1=11
user input the answer:
3+9-1=11
Congratulations, you are right.

```

figure 5

由最终结果可以看出，该实验顺利完成。

### 5.4 对减少火柴棒进行测试

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
3
2
1
2
The question is below: 465-134=334
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
One possible answer of the question is below:
465-131=334
user input the answer:
465-131=334
Congratulations, you are right.

```

figure 6

由最终结果可以看出，该实验顺利完成。

## 5.5 对移动火柴棒进行控制

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
2
2
2
2
2
The question is below: 85-12=19
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
One possible answer of the question is below:
86-72=14
user input the answer:
86-72=14
Congratulations, you are right.

```

figure 7

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
3
2
2
1
The question is below: 973+236=743
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
One possible answer of the question is below:
979-236=743
user input the answer:
979-236=743
Congratulations, you are right.

```

figure 8

由最终结果可以看出，该实验顺利完成。

## 5.6 对用户输入的答案进行控制

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
2
2
2
2
The question is below: 06+18=763
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
85+78=163
user input the answer:

The answer is as follows:
85+78=163

```

figure 9 直接按 Enter

由最终结果可以看出，该实验顺利完成。

## 5.7 测试增加 10 根火柴棒

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
3
3
0
10
The question is below: 525-101+153=111
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
525-307+753=971
user input the answer:
525-307+753=971
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :

```

figure 10

由最终结果可以看出，该实验顺利完成。

## 5.8 测试移动 10 根火柴棒

```

Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :
3
3
2
10
The question is below: 758+828+875=411
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
One possible answer of the question is below:
765+322-906=181
user input the answer:
765+322-906=181
Congratulations, you are right.
Input maxDigit(1-3) operators(2-3) mode(0-2) and bars(0-10) and :

```

figure 11

由最终结果可以看出，该实验顺利完成。

## 六、讨论、心得

此次实验是第一个 Java 工程项目，代码量在 600 行左右，对 Java 的基本语法进行了掌握。有以下三点感受。

第一，scanner 只需要在最后 close，而不是每一次 new 一个 Scanner 就要 close 一次，这样会阻断 scanner 的流。Eclipse 的提示让我纠结了很久，在这个地方出现了长时间的 debug。

第二，要提前想好代码的结构。此次的代码并不是最优的，代码中的方法有几个是可以合并的，代码量可以减少近二百行。

第三，算法思想很重要。我一开始也想过通过暴力枚举的方法来生成题目，但是这样的效率可能更低，所以选用了七段数码管的思想，对每个位进行随机。

通过此次实验，也收获颇多，以后要重点学习 Java 的进阶思想了。