实验二 Java 程序设计语言基础

学生姓名：黄晨箬　　学　　号：6109119066　　专业班级：计算机 193 班

实验类型：■ 验证 □ 综合 □ 设计 □ 创新　实验日期：2021.4.18　实验成绩：

一、实验名称

Java 程序设计语言基础

二、实验目的

1、了解Java 的数据类型；

2、掌握各种变量的声明方式；

3、理解运算符的优先级；

4、掌握Java 基本数据类型、运算符与表达式的使用方法；

5、理解Java 程序语法结构，掌握顺序结构、选择结构和循环结构语法的程序设计方法；

6、通过以上内容，掌握Java 语言的编程规则。

三、实验内容

采用记事本或集成开发环境编写Java语言源程序，完成下列习题任务的运行及调试。

习题1、读入一个浮点数值，将其转换为中文金额的大写方式，如123.45,转换为:壹佰贰拾叁元肆角伍分，并实现：

（1）当金额为整数时，只表示整数部分，省略小数部分，并添加“整”字。例如，123表示为:壹佰贰拾叁元整；

（2）当金额中含有连续的0时，只需写一个“零”即可，例如， 10005表示为：壹万零伍元整；

（3）10的表示方式，例如，110元表示为：壹佰壹拾元整，而10则表示为：拾元整。

提示：将字符串型转换为浮点型可以用Float.parseFloat(s)函数转换。

要求：1、能正确的进行数据转换；

2、能在输入数据错误的情况下给出提示。

3、给出下列输入数据对程序进行测试：

1）123

2）123．11

3）123．10

4）0123．11

正确结果：

1）壹佰二十三元

2）壹佰二十三元一角一分

3）壹佰二十三元一角

4）壹佰二十三元一角一分

问题：本实验中的测试数据前缀0和后缀0怎样处理比较好？

习题2-6：完成课本下列章节的编程题内容。

习题2：　第三章 96页　3.15 （彩票程序）

习题3-4：第四章 133页 4.17 （一个月的天数）

133页 4.18 （学生的专业和年级）

习题5-6：第五章 169页 5.17 （显示金字塔）

171页 5.29 （显示日历）

四、实验仪器设备及耗材

1、PC微机；

2、DOS操作系统或 Windows 操作系统；

3、Eclipse程序集成环境。

五、实验步骤

1、根据题目要求，画出程序流程图；

2、给出习题X程序的java数据结构；

3、编写习题X程序源代码；

4、调试程序：

5、给出一些测试数据，检查输出结果。

六、实验数据及处理结果

习题一：

[源程序]

```java
        int i = -1, w, intFee;
        double money = 0;
        String outtxt = "";//用来记录分析结果
        String[] digits = {"分", "角", "元", "拾", "佰", "仟", "萬", "拾萬", "佰萬", "仟萬", "亿"};
        String[] num = {"零", "壹", "贰", "叁", "肆", "伍", "陆", "柒", "捌", "玖"};
        boolean method;//记录输入的浮点数是否为整数，整数记为true

        try {//执行可能发生异常的语句
            Scanner reader = new Scanner(System.in);
            System.out.println("请输入需要转换的数值");
            money = reader.nextDouble();
            if (money % 1 == 0) {//判断浮点数是否为整数，整数记为true
                method = true;
                outtxt = "整" + outtxt;
                i = 1;
            } else {
                method = false;
                i = -1;
            }
```

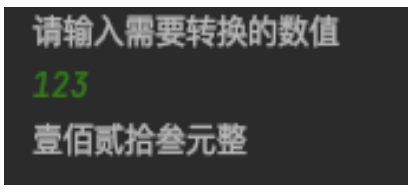
```java
            //如果为整数，则直接转换为int做除法，否则记录两位小数（角和分）
            if (method) {
                intFee = (int) money;
            } else {
                intFee = (int) (money * 100);
            }

            //循环判断并记录应该输出的值
            //如果为中间段有0且在个位，则添加一个元
            //中间段有0且不在个位，不添加0

            while (intFee != 0) {
                w = intFee % 10;
                i += 1;
                if (w != 0) {
                    outtxt = num[w] + digits[i] + outtxt;
                    intFee /= 10;
                } else if (w == 0 && i == 2) {
                    outtxt = digits[i] + outtxt;
                    intFee /= 10;
                } else {
                    intFee /= 10;
                }
            }
            System.out.println(outtxt);
        }
```

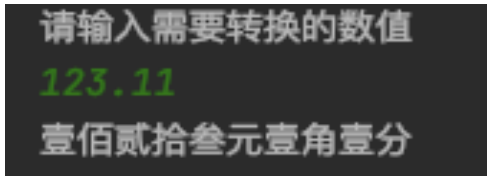
```java
            //预防错误情况
            catch (NumberFormatException e) {
                outtxt = "请输入正确的数额!";
                System.out.println(outtxt);
            }
    }

            }
```

[设计思路或算法]

将各个单位放进数组里，然后在对输入的金额进行判断，对整数与小数进行拆分，根据结果从数组里取相对应的字从而实现中文转换。

[结果及截图]





[分析]
正常运行。

习题二：

[源程序]

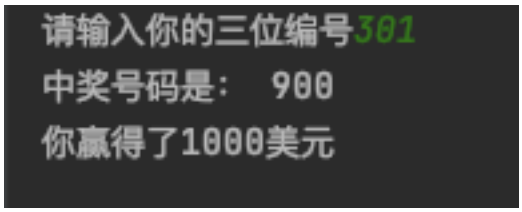
```
int s = (int)(Math.random() * 900 + 100);
System.out.print("请输入你的三位编号");
Scanner input = new Scanner(System.in);
int id = input.nextInt();
if(id < 100 || id > 999)
{
    System.out.println("error input");
    return;
}

//生成中奖号码
int s_001 = s % 10;
int s_010 = s / 10 % 10;
int s_100 = s /100;
int temp = -1;
if(s_100 > s_010){
    temp = s_100;
    s_100 = s_010;
    s_100 = temp;
}
if(s_100 > s_001){
    temp = s_100;
    s_100 = s_010;
    s_100 = temp;
}
```

```
    temp = s_010;
    s_010 = s_001;
    s_001 = s_010;
}
System.out.println("中奖号码是:  " + s_100 + s_010 + s_001);

int id_001 = id % 10;
int id_010 = id / 10 % 10;
int id_100 = id /100;
if(id_100 > id_010){
    temp = id_100;
    id_100 = id_010;
    id_010 = temp;
}
if(id_100 > id_001){
    temp = id_100;
    id_100 = id_010;
    id_010 = temp;
}
if(id_010 > id_001){
    temp = id_010;
    id_010 = id_001;
    id_001 = id_010;
}
```

```java
if(s == id){
    System.out.println("你赢得了10000美元");
}
else if(s_001 == id_001 && s_010 == id_010 && s_100 == id_100)
    System.out.println("你赢得了3000美元");
else if(s_001 == id_001 || s_001 == id_010 || s_001 == id_100 || s_010 == id_001 || s_010 == id_010 || s_010 == id_100 |
    System.out.println("你赢得了1000美元");
else
    System.out.println("没有中奖哦");
```

[设计思路或算法]
省略

[结果及截图]



[分析]
先生成中奖号码的各个数字，然后对他们进行排序将各个位的数字放进对应的变量里。然后读取用户输入的数字，与中奖号码的变量逐一匹配，最后判断是否中奖/中奖金额。

习题三：
[源程序]

```java
//判断是否为闰年
public static int leap(int year) {
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        return 29;
    } else return 28;
}

public static void main(String[] args){
    int days = 0;
    //分别记录用户输入的年月
    System.out.print("Enter a year:");
    Scanner input = new Scanner(System.in);
    int year = input.nextInt();

    System.out.print("Enter a month:");
    Scanner Scan = new Scanner(System.in);
    String month = Scan.next();
```

```
    switch (month) {
        case "Apr" :

        case "Jun" :

        case "Sep" :

        case "Sept" :

        case "Nov" :
            days = 30;
            break;

        case "Jan" :

        case "Mar" :

        case "Jul" :

        case "Aug" :

        case "Oct" :

        case "Dec" :
            days = 31;
            break;
```

```
            case "Feb" :
                days = leap(year);
                break;
        }
        if (days == 0) {
            System.out.println(month + " is not a correct month name");
        } else {
            System.out.println(month + " " + year + " has " + days + " days");
        }
```

[设计思路或算法]

```
/*
    按照平年来计算的话，一年中的1，3，5，7，8，10，12月为31天
    4，6，9，11为30天
    2月为28天

    和平年不同的是，闰年的2月为29天

    所以可以首先判断月份是否是2月，如果不是，则直接判断得出是否为31天，如果是，则对年份进行判断
*/
```

[结果及截图]

```
Enter a year:2001
Enter a month:Jan
Jan 2001 has 31 days
```

```
Enter a year:2016
Enter a month:jan
jan is not a correct month name
```

[分析]
正常运行。

习题四：
[源程序]

```
//录入用户输入的字符
Scanner scanner = new Scanner(System.in);
System.out.println("Enter two characters: ");
String str = scanner.nextLine();
scanner.close();
char[] Arr = str.toCharArray();

String major = String.valueOf(Arr[0]);
switch (major) {
    case "M" :
        major = "Mathematics";
        break;

    case "C" :
        major = "Computer Science";
        break;

    case "I" :
        major = "Information Technology";
        break;
}
```

```
String grade = String.valueOf(Arr[1]);
switch (grade) {
    case "1" :
        grade = " Freshman";
        break;

    case "2" :
        grade = " Sophomore";
        break;

    case "3" :
        grade = " Junior";
        break;

    case "4" :
        grade = " Senior";
        break;
}

if (major.equals(String.valueOf(Arr[0]))) {
    System.out.println("Invalid input");
}else if (grade.equals(String.valueOf(Arr[1]))) {
    System.out.println("Invalid Input");
}else {
    System.out.println(major + grade);
}
```

[设计思路或算法]

```
/*
    首先录入用户输入的字母，然后将输入的字符串改为数组的形式存储。
    数组的第一个元素代表专业，对其进行判断得出相对应的专业信息
    数组的第二个元素代表年级，对其进行判断得出相对应的年级信息
    将以上内容输出即可
*/
```

[结果及截图]

```
Enter two characters:
M1
Mathematics Freshman
```

```
Enter two characters:
C3
Computer Science Junior
```

```
Enter two characters:
T3
Invalid input
```

[分析]

正常运行。

习题五：

[源程序]

```
System.out.print("Enter the number of the lines:");
Scanner input = new Scanner(System.in);
int n = input.nextInt();
int m = 2*n+1;

int i,j;
int[][] arr = new int[n][m];
for (i = 0;i < n; i++)
    for (j = n-i;j <= n+i; j++)
        if (n-j > 0){
            arr[i][j] = (n-j) + 1;
        }
        else {
            arr[i][j] = -(n-j) + 1;
        }

for (i = 0;i < n;i++) {
    for (j = 0; j < m; j++) {
        if (arr[i][j] == 0) {
            System.out.print(" ");
        }else {
            System.out.print(arr[i][j]);
        }
    }
    System.out.println();
}
```

[设计思路或算法]

```
每一行有2*n+1个数字(从0计数)
一共有n行
使用二维数组遍历输出所需要的结构
首先定义一个数组[n][2*n+1]
将各个数据存入数组的对应位置
每个位置的大小应该是总行数与当前行数的差的差加一
金字塔的数据呈一个类似镜面，所以我们只需要考虑一边，另一边只需要对其取绝对值即可
由于每一行的第一个数总是和当前的行数相同，于是只要用行数表示列数，再递增即可确定数的位置
最后遍历输出
```

[结果及截图]

```
Enter the number of the lines:7
      1
     212
    32123
   4321234
  543212345
 65432123456
7654321234567
```

[分析]

正常运行。

习题六：

[源程序]

```
//判断闰年
public static  int leap(int year){
    if ((year % 4 == 0 && year % 100 != 0) || year % 400 == 0) {
        return 1;
    } else return 0;
}


//通过（年份总和+基准天）%7计算年份的第一天
public static int  firstDayOfYear(int year) {
    int baseYear = 2000;
    int firstDay = 6;
    int i = 2000;
    int total = 0;
    for (i = baseYear; i < year; i++)
        total = total + 365 + leap(i);
    return (total + firstDay) % 7;
}

//判断月份的第一天
public static int firstDayOfMonth(int year,int month, int firstYear){
    int[][] mon = {{0,31,28,31,30,31,30,31,31,30,31,30,31},{0,31,29,31,30,31,30,31,31,30,31,30,31}};
    int total = 0;
    int i = 0;
    for (i = 1; i < month; i++)
        total = total + mon[leap(year)][i];
    return (total + firstYear)%7;
}
```

```java
//对日历进行展示
public static void show(int year,int month,int firstMonth){
    int[][] mon = {{0,31,28,31,30,31,30,31,31,30,31,30,31},{0,31,29,31,30,31,30,31,31,30,31,30,31}};
    int i = 0;
    System.out.println("\t"+ month+ " "+ year);
    System.out.println("————————————————");
    System.out.println("Mon  Tue  Wen  Thu  Fri  Sat  Sun");
    for(i=0; i<firstMonth-1; i++)
        System.out.print("      ");
    for (i=1; i<=mon[leap(year)][month]; i++) {
        System.out.print(String.format("%-5d", i));
        if ((firstMonth - 1 + i) % 7 == 0) System.out.println();
    }
    System.out.println();
}

public static void main(String[] args) {

    System.out.print("输入年份");
    Scanner input = new Scanner(System.in);
    int year = input.nextInt();

    System.out.print("输入月份");
    Scanner Input = new Scanner(System.in);
    int month = Input.nextInt();

    int firstYear = firstDayOfYear(year);
    int firstMonth = firstDayOfMonth(year,month,firstYear);
    show(year,month,firstMonth);
}
```

[设计思路或算法]

判断我们想打印某年某月的日历，我们就得知道这个月是从星期几开始，且这个月有多少天。要想知道这个月是从星期几开始的，就得知道上一个月的最后一天是星期几，要想知道上一个月的最后一天是星期几，就得知道上一个月是星期几开始的，因此，指定一个基准年（我选择的是2000年），然后定基准日（也就是基准年第一天星期几）。然后就能通过基准计算某年某月从哪天开始。

[结果及截图]

```
输入年份2013
输入月份1
    1 2013
————————————————
Mon  Tue  Wen  Thu  Fri  Sat  Sun
      1    2    3    4    5    6
7    8    9    10   11   12   13
14   15   16   17   18   19   20
21   22   23   24   25   26   27
28   29   30   31
```

```
输入年份2013
输入月份12
    12 2013
————————————————
Mon  Tue  Wen  Thu  Fri  Sat  Sun
1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30   31
```

[分析]

发现2013.12没有正常显示，调试后发现，当该月的第一天是星期天时，显示空格的语句不能正常执行（此时i<-1）：

```
for(i=0; i<firstMonth-1; i++) {
    System.out.print("      ");
}
```

针对该种情况进行修改：

```
if(firstMonth == 0) {
    System.out.print("                                    ");
}
```

再进行调试：

```
输入年份2013
输入月份12
    12 2013
————————————————
Mon  Tue  Wen  Thu  Fri  Sat  Sun
                             1
2    3    4    5    6    7    8
9    10   11   12   13   14   15
16   17   18   19   20   21   22
23   24   25   26   27   28   29
30   31
```

正常运行。

七、思考题

1、Java语言中的循环语句与其他语言中的有何区别？

Java的循环语句使用方法和C语言类似，但Java中判断条件只能为布尔表达式，只

有在括号中的布尔表达式为真的时候，循环才会继续。

2、如何获取基本数据类型byte,short,int,long,float,double表示的最大、最小值？

要获取最大值，只需要添加.MAX_VALUE,要获取最小值，只需要添加.MIN_VALUE。

以byte为例，获取最大值使用Byte.MAX_VALUE;获取最小值使用Byte.MIN_VALUE。

八、实验总结及体会

通过本次实验，学习巩固了循环语句的使用，了解了诸多数据类型，熟悉各个运算符和使用方法，熟练了对 Java 程序设计的使用。