实验1

练习1

题目

求$\sqrt{6(\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\cdots+\frac{1}{100^2})}$的近似值

解析

简单累加即可。

程序使用了从小到大的累加方式，以提高计算精度。（与计算机中的浮点数保存方式(IEEE754)有关，不展开）

代码

```
public class Exercise1 {

    public static void main(String[] args) {
        double result = 0;
        for (int i = 100; i > 0; i--) {
            result += 1.0 / i * i;
        }
        result = Math.sqrt(6 * result);
        System.out.println(result);
    }

}
```

输入

```
无
```

输出

```
24.49489742783178
```

练习2

题目

求$4(1-\frac{1}{3}+\frac{1}{5}-\frac{1}{7}+\cdots+\frac{1}{19997}-\frac{1}{19999})$的近似值

解析

循环累加，在第一个循环中进行加和运算，在第二个循环中进行相减运算。

同样采用从小到大的加和方式，以提高计算精度。

代码

```
public class Exercise2 {

    public static void main(String[] args) {
        double result = 0;
        for (int i = 19997; i > 0; i -= 4) {
            result += 1.0 / i;
        }
        for (int i = 19999; i > 2; i -= 4) {
            result -= 1.0 / i;
        }
        result *= 4;
        System.out.println(result);
    }

}
```

输入

```
无
```

输出

```
3.1414926535900536
```

练习3

题目

求$2(\frac{2}{1}\times\frac{3}{2}\times\frac{4}{3}\times\frac{4}{5}\times\frac{6}{5}\times\frac{6}{7}\times\frac{8}{7}\times\frac{8}{9}\times\cdots\times\frac{19998}{19997}\times\frac{19998}{19999})$的近似值。

解析

同理，仅需在练习2的基础上稍作改动。

代码

```
public class Exercise3 {
    public static void main(String[] args) {
        double result = 1;
        for (int i = 19999; i > 2; i -= 2) {
            result *= i - 1;
            result /= i;
        }
        for (int i = 19997; i > 0; i -= 2) {
            result *= i + 1;
            result /= i;
        }
        result *= 2;
        System.out.println(result);
    }
}
```

输入

```
无
```

输出

```
3.1415141108281435
```

练习4

题目

求2^{1000}的末尾3位数字。

解析

初值为1，仅需每次乘以2后取后三位即可，重复1000次。

代码

```
public class Exercise4 {

    public static void main(String[] args) {
        int result = 1;
        for (int i = 0; i < 1000; i++) {
            result *= 2;
            result %= 1000;
        }
        System.out.println(result);
    }

}
```

输入

```
无
```

输出

```
376
```

练习5

题目

在一个整数序列中，有一个整数的出现次数超过了一半。求该整数。说明：数据自定。

解析

程序通过调用Java内置的数据结构HashMap实现。下面给出简要步骤：

1. 创建变量名为count的HashMap，从Integer映射到Integer
2. 遍历输入数组arr，若count中不存在键arr[i]，则添加键值对arr[i] -> 1，否则使arr[i]对应的值加1
3. 当arr[i]对应的值大于arr长度的一半时，中止循环，打印arr[i]

由于使用了散列表，因此程序的时间复杂度应当很低。如果不允许使用HashMap，也可使用桶排序的思想或直接逐个搜索，甚至使用两个数组简单代替HashMap，但这样会提高时间复杂度，因此不附加使用这些方法的代码。

代码

```
import java.util.HashMap;

public class Exercise5 {

    public static void main(String[] args) {
        int[] arr = { ... }; //arr的具体值将在下面的“输入”中给出
        HashMap<Integer, Integer> count = new HashMap<Integer, Integer>();
        for (int i = 0; i < arr.length; i++) {
            if (count.get(arr[i]) == null) {
                count.put(arr[i], 1);
            } else {
                count.put(arr[i], count.get(arr[i])+1);
            }
            if (count.get(arr[i]) > arr.length / 2) {
                System.out.println(arr[i]);
                break;
            }
        }
    }

}
```

注意到代码中多次调用了count.get()方法，实际上可以对其进行优化，但为代码清晰起见，不做过多优化。

事实上，JVM本身会对这样的代码在编译时进行优化，无需过多考虑这样的问题。

输入

```
{30, 32, 32, 66,  7, 32, 32, 32, 86, 32,
 32, 40, 84, 11, 88, 32, 32, 32, 32,  5,
 81, 79, 32, 32, 77,  8, 32, 32, 48, 32,
 24, 32, 32, 37, 18, 32, 17, 32, 23,  6,
 62, 32, 59, 95, 32, 32,  5, 45, 29, 32,
 32, 97, 32, 51, 75, 32, 32, 32, 32,  6,
 48, 32, 32, 16, 32, 32, 64, 32, 53, 32,
 68, 32, 32, 32, 32, 32, 49, 69, 32, 32,
  5, 50, 98, 87, 32, 99, 32,  9, 69, 32,
 62, 32, 29, 32, 32, 32, 32, 32, 48, 32}
```

输出

```
32
```

心得体会

1. 对浮点数而言，从小到大进行累加会得到更好的精度。
2. 散列表在很多情况下能实现低时间复杂度的算法。
3. 对练习5的考虑还不够周密，应当存在使用更简单数据结构的低时间复杂度算法。