

中北大学软件学院

实 验 报 告

专 业：	软件工程
课程名称：	面向对象程序设计
班 级：	22130403
学 号：	2213040340
姓 名：	张恒瑜
辅导教师：	李华玲

2021 年 9 月 制

成绩：_____

实验时间	2023 年 10 月 2 日 8 时至 10 时	学时数	2 学时
1. 实验名称 类与对象（1）			
2. 实验目的 （1）学会类的实现、类的使用和用 UML 类图描述类之间的关系，能够对实际的案例进行分析和建模，并画出类图。 （2）能够根据类图进行编码实现。 （3）理解构造方法的概念和重载的概念，能在程序设计中灵活应用。 （4）巩固数组在面向对象程序设计中的运用；			
3. 训练知识点集群 （1）类的语法（类的成员变量和成员方法、构造方法、成员变量的 getter 、 setter 方法等） （2）类图（单个类的类图和类之间的关系） （3）使用数组构造线性表			
4. 实验内容 （1）计算某长方体或正方体盒子的体积 （2）课程融合之线性表 ➤ 定义一个类 MyArrayList ，用数组模拟实现数据结构中线性表的基本功能，该类中最少包含的属性有： data （ String[] ）、 size （ int ）、 CAPACITY （ int ）； ➤ 该类中方法最少包括线性表的初始化、插入、查找、遍历线性表、删除、返回线性表长度、线性表判空、清空线性表、替换、获取元素、判断线性表中是否包含某个元素、反向查找、末尾插入元素； ➤ 定义一个主类，在主方法中创建线性表类对象并调用上述方法、完成功能演示。			

5. 根据问题抽象出类图

Box
-size: float -Height:float -Width:float -Length:float
+getVolume(size);float +getVolume(height,width,length):float

MyArrayList
-size:int -CAPACITY:(int) =100 -data:(String[CAPACITY])
+MyArrayList() +MyArrayList(String[]) +Insert(String):void +Insert(int,String):void +showAll():void +deleteList(int):void +size():int +isEmpty():boolean +clear():void +takeplaceElem (int , string):void +gainElem(int):float +Contain(float):boolean +indexOf(String):int +lastIndexOf(String):int

5. 实验源代码

(1)



TestBoxVolume.java

```
import java.util.Scanner;
```

```

class Box {
    public float getVolume(float size) {
        return size*size*size;
    }
    public float getVolume(float height,float width,float length) {
        return height*width*length;
    }
}

public class TestBoxVolume {
    public static void main (String[] args) {
        Scanner sc = new Scanner(System.in);
        Box box = new Box();
        while (true) {

System.out.println("*****");

            System.out.println("***** 请 选 择 你 要 进
行的 计 算 *****");

            System.out.println("*****1.正方体的运
算 *****");

System.out.println("*****2. 长 方 体 的 运 算
*****");

System.out.println("*****3. 退 出 主 程 序
*****");

System.out.println("*****");
                int num = sc.nextInt();
                switch (num) {
                    case 1 -> {

                        System.out.println("请输入正方体的
边长:");

                        float size = sc.nextFloat();
                        System.out.println("正方体的体积是:

```

```
" + box.getVolume(size));  
        }  
        case 2 -> {  
  
            System.out.println("请输入长方体的  
长：");  
  
            float height = sc.nextFloat();  
            System.out.println("请输入长方体的  
宽：");  
  
            float width = sc.nextFloat();  
            System.out.println("请输入长方体的  
高：");  
  
            float length = sc.nextFloat();  
            System.out.println("长方体的体积是：  
" + box.getVolume(height,width,length));  
        }  
        case 3 -> {  
            return;  
        }  
    }  
}  
}
```

(2)



TestMyArrayList.java

```
import java.util.Scanner;
```

```
class MyArrayList {
```

```
private int size = 0;
private final int CAPACITY = 100;
private String[] data;

public MyArrayList () { // 初始化线性表
    data = new String[CAPACITY];
}

public MyArrayList (String[] data) {
    super(); // 继承了父类
    this.data = data;
}

public void Insert(int index, String s) { // 在某一位置插入
    for(int i = size-1; i >= index; i--) {
        data[i+1] = data[i];
    }
    data[index] = s;
    size++;
}

public void Insert(String s) { // 在末尾插入
    Insert(size, s);
}

public void showAll() { // 遍历后打印所有元素
    System.out.print("[");
    for(int i = 0; i < size; i++) {
        System.out.print(data[i]);
        if(i < size-1) {
            System.out.print(",");
        }
    }
    System.out.println("]");
}

public void deleteList(int index) { // 删除线性表中某个下标元
```

素

```
        for(int i = index-1; i < size; i++) {  
            data[i] = data[i+1];  
        }  
        size--;  
        System.out.println("已删除");  
    }
```

```
    public int size() { //线性表的大小  
        return size;  
    }
```

```
    public boolean isEmpty() { //判断线性表是否为空  
        return size == 0;  
    }
```

```
    public void clear() { //清除线性表  
        size = 0;  
    }
```

中某个下标中的元素

```
        data[index-1] = s;  
    }
```

素

```
        return data[index-1];  
    }
```

某个元素

```
    public boolean Contain(String s) { //判断线性表中是否包括
```

```

        for(int i = 0; i < size; i++) {
            if(data[i].equals(s)) {
                return true;
            }
        }
        return false;
    }

    public int indexOf(String s) { //查找某个元素的下标（正向）
        for(int i = 0; i <= size; i++) {
            if (data[i].equals(s)) {
                return i+1;
            }
        }
        return -1;
    }

    public int lastIndexOf(String s) { //查找某个元素的下标（反
    向）
        for (int i = size-1; i > 0; i--) {
            if(data[i].equals(s)) {
                return i+1;
            }
        }
        return -1;
    }
}

public class TestMyArrayList {
    public static void main (String[] args) {
        Scanner sc1 = new Scanner(System.in);
        MyArrayList list = new MyArrayList();
        while(true) {

System.out.println("*****");

            System.out.println("*****请输入你想进行
            的操作*****");

            System.out.println("*****1.初始化线性表

```



```
*****");  
        System.out.println("*****2.插入元素到线性表*****");  
  
        System.out.println("*****3.陈列所创建的线性表*****");  
  
        System.out.println("*****4.删除线性表中的元素*****");  
  
        System.out.println("*****5.显示线性表的大小*****");  
  
        System.out.println("*****6.判断线性表是否为空*****");  
  
        System.out.println("*****7.替换线性表中的元素*****");  
  
        System.out.println("*****8.给出线性表中的元素*****");  
  
        System.out.println("*****9.判断线性表中包含的元素*****");  
  
        System.out.println("*****10.正向查找线性表中的元素*****");  
  
        System.out.println("*****11.反向查找线性表中的元素*****");  
  
        System.out.println("*****12.清除线性表
```

```

中所有元素*****");

        System.out.println("*****13. 退出主程序
*****");

System.out.println("*****");
        int choose = sc1.nextInt();
        switch (choose) {
            case 1 -> {

                System.out.println("请输入你想要
创建线性表的大小：");

                int    size    =
sc1.nextInt();

                for(int i = 1; i <= size ; i++ ) {

                    System.out.println("请输入第"
+ i + "个元素");

                    String elem = sc1.next();
                    list.Insert(elem);

                }
            }
            case 2 -> {

                System.out.println("请输入你要插
入的位置：");

                int location = sc1.nextInt();

                System.out.println("请输入你要插
入的元素：");

                String elem = sc1.next();

                list.Insert(location,elem);

            }
            case 3 -> {

                list.showAll();

            }
            case 4 -> {

```

```

        System.out.println("请输入你要删除线性表中第几个元素：");

        int index = sc1.nextInt();
        list.deleteList(index);
    }
    case 5 -> {
        System.out.println("该线性表的大小是：" + list.size());
    }
    case 6 -> {
        System.out.println(list.isEmpty());
    }
    case 7 -> {
        System.out.println("请输入你要替换元素的位置：");

        int location = sc1.nextInt();
        System.out.println("请输入你要替换的元素：");

        String elem = sc1.next();
        list.takeplaceElem(location,elem);
    }
    case 8 -> {
        System.out.println("请输入你要查找的位置：");

        int index =
sc1.nextInt();

System.out.println(list.gainElem(index));
    }
    case 9 -> {
        System.out.println("请输入一个元素：");

```

```
String elem = sc1.next();

System.out.println(list.Contains(elem));
    }
    case 10 -> {
        System.out.println("请输入一个元素：");

        String elem = sc1.next();
        System.out.println("这个元素的位置是：" + list.indexOf(elem));
    }
    case 11 -> {
        System.out.println("请输入一个元素：");

        String elem = sc1.next();
        System.out.println("这个元素的位置是：" + list.lastIndexOf(elem));
    }
    case 12 -> {
        list.clear();
    }
    case 13 -> {
        return;
    }
}

}
```

7.实验运行结果截图及心得体会

```
D:\zhy0340_java\chap03>javac TestMyArrayList.java
```

```
D:\zhy0340_java\chap03>java TestMyArrayList
```

```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

```
1
```

```
请输入你想要创建线性表的大小：
```

```
3
```

```
请输入第1个元素
```

```
zhang
```

```
请输入第2个元素
```

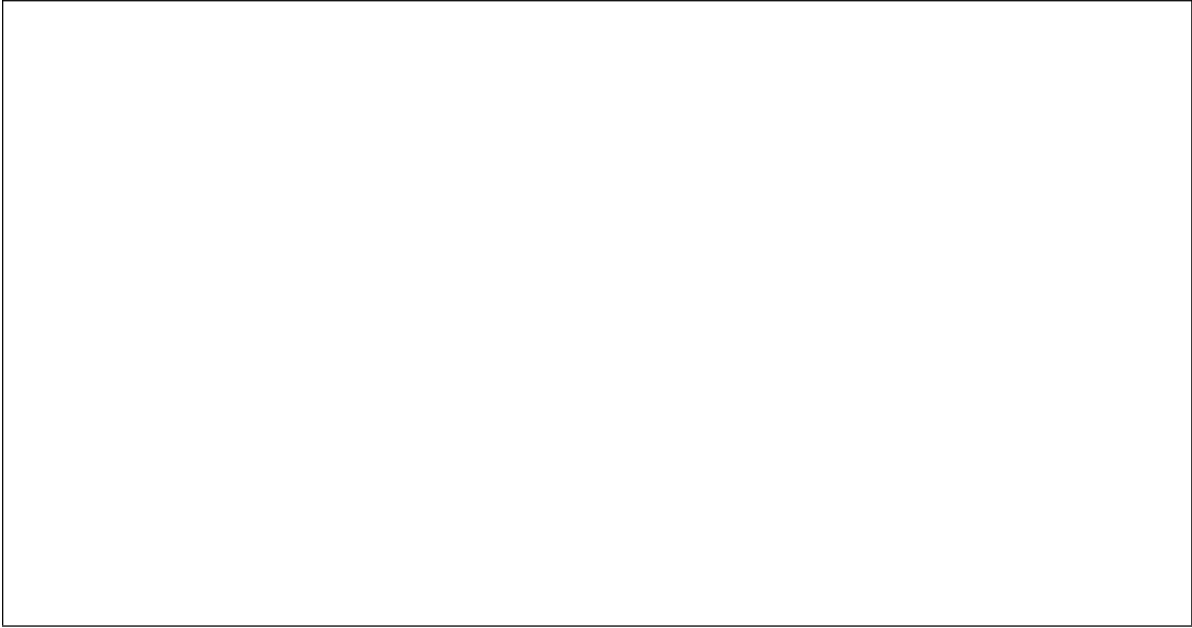
```
heng
```

```
请输入第3个元素
```

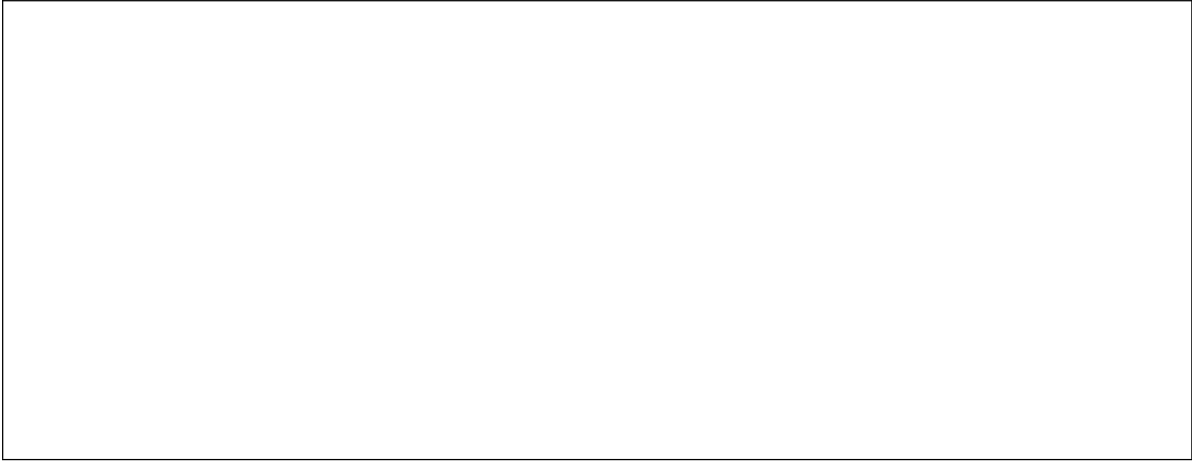
```
yu
```

```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

```
2
```



```
2
请输入你要插入的位置：
3
请输入你要插入的元素：
shuai
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
3
[zhang,heng,yu,shuai]
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```




```
4
请输入你要删除线性表中第几个元素：
4
已删除
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
3
[zhang,heng,yu]
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
5
该线性表的大小是：3
```



6

false

```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

7

请输入你要替换元素的位置：

3

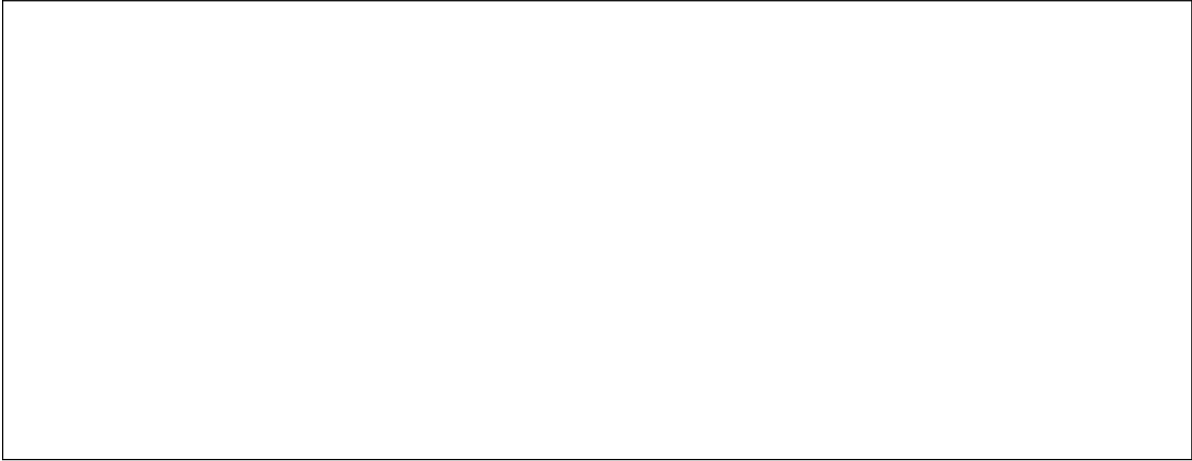
请输入你要替换的元素：

瑜shuai

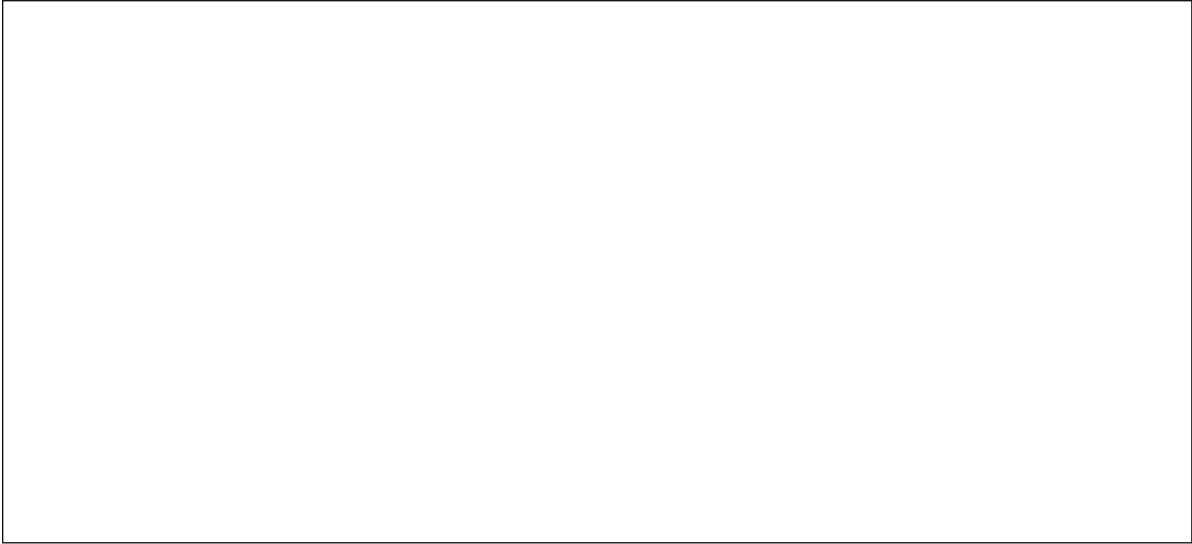
```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

3

[zhang,heng,瑜shuai]



```
8
请输入你要查找的位置：
2
heng
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
9
请输入一个元素：
zhang
true
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
10
请输入一个元素：
zhang
这个元素的位置是： 1
*****
```



11

请输入一个元素：

heng

这个元素的位置是：2

```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

12

```
*****
*****请输入你想进行的操作*****
*****1.初始化线性表*****
*****2.插入元素到线性表*****
*****3.陈列所创建的线性表*****
*****4.删除线性表中的元素*****
*****5.显示线性表的大小*****
*****6.判断线性表是否为空*****
*****7.替换线性表中的元素*****
*****8.给出线性表中的元素*****
*****9.判断线性表中包含的元素*****
*****10.正向查找线性表中的元素*****
*****11.反向查找线性表中的元素*****
*****12.清除线性表中所有元素*****
*****13.退出主程序*****
*****
```

3

[]


```
D:\zhy0340_java\chap03>javac TestBoxVolume.java
```

```
D:\zhy0340_java\chap03>java TestBoxVolume
```

```
*****  
*****请选择你要进行的计算*****  
*****1.正方体的运算 *****  
*****2.长方体的运算 *****  
*****3.退出主程序 *****  
*****
```

```
1
```

```
请输入正方体的边长:
```

```
13
```

```
正方体的体积是: 2197.0
```

```
*****  
*****请选择你要进行的计算*****  
*****1.正方体的运算 *****  
*****2.长方体的运算 *****  
*****3.退出主程序 *****  
*****
```

```
2
```

```
请输入长方体的长:
```

```
12
```

```
请输入长方体的宽:
```

```
13
```

```
请输入长方体的高:
```

```
15
```

```
长方体的体积是: 2340.0
```

```
*****  
*****请选择你要进行的计算*****  
*****1.正方体的运算 *****  
*****2.长方体的运算 *****  
*****3.退出主程序 *****  
*****
```

```
3
```

```
D:\zhy0340_java\chap03>|
```

```
D:\zhy0340_java\chap03>javac TestBoxVolume.java
```

```
D:\zhy0340_java\chap03>java TestBoxVolume
```

```
*****  
*****请选择你要进行的计算*****  
*****1. 正方体的运算 *****  
*****2. 长方体的运算 *****  
*****3. 退出主程序 *****  
*****
```

```
1
```

```
请输入正方体的边长:
```

```
13
```

```
正方体的体积是: 2197.0
```

```
*****  
*****请选择你要进行的计算*****  
*****1. 正方体的运算 *****  
*****2. 长方体的运算 *****  
*****3. 退出主程序 *****  
*****
```

```
2
```

```
请输入长方体的长:
```

```
12
```

```
请输入长方体的宽:
```

```
13
```

```
请输入长方体的高:
```

```
15
```

```
长方体的体积是: 2340.0
```

```
*****  
*****请选择你要进行的计算*****  
*****1. 正方体的运算 *****  
*****2. 长方体的运算 *****  
*****3. 退出主程序 *****  
*****
```

```
3
```

```
D:\zhy0340_java\chap03>|
```

心得体会：比预期的时间用的要长，学会了运用一些函数，比如 `math` 中的数学函数和 `equals` 等。了解了线性表的工作原理，感觉收获挺多，再接再厉