

# 面向对象程序设计 实验指导书

肖磊、宋鸿陟、彭红星  
2012-10

## 目录

实验 1 猜数字游戏 .....	1
实验 2 发牌游戏 .....	2
实验 3 实现一个圆类 .....	3
实验 4 Java 源代码分析程序 .....	4
实验 5 银行账户存取款业务 .....	5
实验 6 深克隆的 House 类 .....	6
实验 7 计算多个二维图形的面积之和 .....	7
实验 8 学生成绩分析程序 .....	8

# 实验 1 猜数字游戏

实验类型：验证性

实验目的：掌握 Java 语言数据类型、基本输入输出、控制语句和方法的使用。

实验内容：

编写程序，项目名与类名均为GuessNumberGame。

每次游戏程序随机产生一个0-9之间的整数，要求玩家输入自己猜的数字，并对用户输入数字进行检查，进行如下提示：

- 如果用户猜对了，则提示：恭喜你，猜对了。结束本次游戏。
- 如果用户猜错了，则提示：你猜的数字太（大或小）了。要求用户继续猜。

如果连续3次没有猜对，则提示：游戏失败。

一次游戏结束时，提示用户是否继续新的游戏，果用户选择继续，则开始新一次游戏，否则输出：你共进行了XXX次游戏，成功XXX次，失败XXX次。然后结束程序运行。

提交要求：打包为可以执行的JAR文档，其中要包含源程序文件。

## 实验 2 发牌游戏

实验类型：验证性

实验目的：掌握 Java 数组、方法、类的基本定义。

实验内容：

编写程序，项目名和类名均为 PokerGame。

描述：

共有 1 幅扑克牌，不包括大王和小王共 52 张牌。

可能有  $n$  个人参与扑克游戏， $2 \leq n \leq 6$ 。

程序运行时输入  $n$ ，然后 52 张牌分别依次分发给  $n$  个人。不能整除时，每个人的牌数可以不同，如 3 个人，则第 1 个人 18 张，第 2 个和第 3 个人 17 张牌。

发牌完成后按花色（顺序为黑桃、红心、草花、方块）和牌面大小输出每个人得到的牌。

例如：

输入人数：3

输出如下：

第 1 个人：

黑桃：K 10 5A

红心：10 3 2

草花：K 10 8 6 3 A

方块：Q J 5 2

第 2 个人：

.....

第 3 个人：

.....

要求：

（1）使用数组存放发牌情况。

（2）编写不同方法完成不同功能。

提交要求：打包为可以执行的 JAR 文档，其中要包含源程序文件。

## 实验 3 实现一个圆类

实验类型：验证性

实验目的：掌握 Java 语言类的定义、数据域封装、构造方法

实验内容：

一、定义一个名为 Circle 的圆类，位于 shape 包中。要求如下：

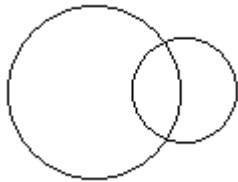
(1) 该类需要描述圆在二维坐标系中的位置，圆心的横、纵坐标值均取整数；描述圆的半径，取双精度类型。

(2) 无参构造方法将圆心坐标设为坐标原点，半径设为 1.0；有参构造方法需要根据参数值设置圆心坐标和半径值。

(3) 提供一个计算圆面积的方法 `double getArea()`。

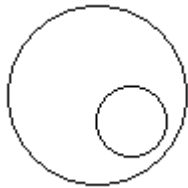
(4) 提供一个判断当前圆对象与参数圆对象是否相交的方法(相交方式见下图)：

`boolean isIntersected(Circle another)`



(5) 提供一个判断当前圆对象是否包含参数圆对象的方法(包含方式见下图)：

`boolean contains(Circle another)`



(6) 提供一个判断当前圆对象与参数圆对象的大小关系的方法(判断依据是半径，半径之差的绝对值小于 0.0001 认为相等；返回负数表示当前圆对象小，0 表示相等，正数表示当前圆对象大)

`int compareTo(Circle another)`

(7) 提供返回当前圆的字符串的方法，格式为：`[Circle(x,y)-radius]`

`String toString()`

二、定义一个包含 main 方法的类 TestCircle，位于 shape 包中。要求如下：

(1) main 方法中调用以下方法。

(2) 编写静态方法 `createArray`，生成位置和大小均随机的 10 个圆对象，置于数组中。

(3) 编写静态方法 `sort` 对这 10 个圆对象进行升序排序，并输出。

(4) 编写静态方法 `computeArea` 计算这 10 个圆中互不相交和包含的圆的面积之和。

提交要求：打包为可以执行的 JAR 文档，其中要包含源程序文件。

## 实验 4 Java 源代码分析程序

实验类型：验证性

实验目的：掌握 Java 语言的字符串处理、文本文件的读写

实验内容：

Java 语言共有 50 个关键字（自行查找相关文档），已知 Java 源程序是以 “.java” 为扩展名的文本文件。可以考虑在 Java 源程序中共有 3 类元素：

〈1〉 代码行，可运行的 Java 源代码；

〈2〉 注释行，3 种注释均可；

〈3〉 空行，既无代码，也无注释；

编写一个类名为 JavaCodeAnalyzer 的类，实现以下功能：

〈1〉 该类可以以一个 Java 源程序文件名创建一个对象，如果文件不存在，或扩展名不是 “.java”，应提示文件错误；

〈2〉 对该源程序文件进行分析其中的：代码行数、注释行数和空行数。说明：行是指以回车结束的。

〈3〉 统计并按从多到少输出出现次数最多的 5 个关键字及其出现次数。

〈4〉 统计分析的结果输出到名为：result.txt 的文本文件中，输出格式如下：

分析 XXXXX.java 源程序的结果：

代码行数：xxx，占 xx.xx%

注释行数：xxx，占 xx.xx%

空白行数：xxx，占 xx.xx%

源程序总行数：xxxx

出现最多的 5 个关键字是：

关键字 1：xx 次

关键字 2：xx 次

关键字 3：xx 次

关键字 4：xx 次

关键字 5：xx 次

分析时间：年-月-日，时-分-秒

〈5〉 测试时要通过以 Java7 的 String 类的源码测试。

提交要求：打包为可以执行的 JAR 文档，其中要包含源程序文件。

## 实验 5 银行账户存取款业务

实验类型：验证性

实验目的：掌握继承、覆盖、多态、对象类型转换

实验内容：

要求编写程序模拟银行账户的存、取款操作。按要求完成以下步骤：

**步骤 1：**编写程序 Account.java，其中定义银行账户类 Account。该类中有账号、姓名、存款余额等数据域，余额默认是 0；有存款、取款、获取当前余额等方法。其中账号为长度为 12 位数字的字符串，姓名为字符串，存款余额为 double。

**步骤 2：**编写名为 CreditAccount 类的信用卡账户类。该类继承自 Account 类，增加一个透支限额(double)数据域，透支限额默认为 1000。同时该类账户取款时允许透支，但不能超过透支限额。

**步骤 3：**编写名为 SavingAccount 的储蓄账户类 SavingAccount。该类继承自 Account 类。该类账户取款时不允许透支。

**步骤 4：**编写名为 Bank 的模拟银行类，其中可以存储多个类型可能是信用卡账户或储蓄账户的对象(可以用数组或 ArrayList 实现)。该类包含以下方法：

- <1>开户：即增加一个新的账户，注意：不允许两个账户的账号相同
- <2>销户：即删除一个已有的账户
- <3>查询账户：根据一个账号，查询有无该账号的账户
- <4>统计目前银行的存款总余额的方法。
- <5>统计目前银行的信用卡账户总透支金额的方法。
- <6>统计目前总账户数。
- <7>统计目前信用卡账户数
- <8>统计目前储蓄卡账户数

**步骤 5：**编写客户端类 Client.java 完成以下功能：

- <1> 编写一个静态方法，创建一个银行对象，并随机生成 10 个账号从 1000 0000 0000 0000 到 1000 0000 0000 0009、类型不同的账户。
- <2> main 方法中模拟几次开户操作。
- <3> main 方法中模拟几次销户操作。
- <4> 模拟几个对指定账号的存款和取款操作。
- <5> 输出银行的总账户数、总余额、总透支数、各类具体账户数。

提示：可以尝试一下范型。

提交要求：打包为可以执行的 JAR 文档，其中要包含源程序文件。

## 实验 6 深克隆的 House 类

实验类型：验证性

实验目的：掌握继承、覆盖、多态、对象类型转换

实验内容：

定义一个深克隆的 House 类。

要求使用如下代码测试时有相应的输出结果：

```
public class Tester{
    public static void main(String[] args)
    {
        House h1 = new House(1, 200.0);
        House h2 = (House)h1.clone();
        System.out.println(h1==h2);          //输出 false
        System.out.println(h1.equals(h2));    //输出 true

        System.out.println(h1.getWhenBuilt()==h2.getWhenBuilt()); //输出 false
        System.out.println(h1.getWhenBuilt().equals(h2.getWhenBuilt())); //输出
true
    }
}
```

提交要求：打包为可以执行的JAR文档，其中要包含源程序文件。



## 实验 7 计算多个二维图形的面积之和

实验类型：验证性

实验目的：掌握接口的定义、接口实现、多态、对象类型转换

实验内容：

假设已经存在如下计算图形面积之和的类 ShapeUtils.java。

```
package shape;

import action.Quadrable;
import java.util.ArrayList;

public class ShapeUtils {

    //计算 ArrayList 中的多个图形的面积之和
    public static double totalArea(Quadrable[] array) {
        double result = 0.0;
        for (Quadrable shape : array) {
            result += shape.getArea(); //getArea 是计算图形面积的方法
        }
        return result;
    }

    //计算数组中的多个图形的面积之和
    public static double totalArea(ArrayList<Quadrable> list) {
        double result = 0.0;
        for (int i = 0; i < list.size(); i++) {
            result += list.get(i).getArea(); //getArea 是计算图形面积的方法
        }
        return result;
    }
}
```

其中：Quadrable 是你需要定义的一个接口，getArea 是该接口中定义的抽象方法。

要求：

<1> 定义接口 Quadrable.java

<2> 定义多个可计算图形面积的图形类（至少三个：Circle、Rectangle、Triangle）。

每个类均需要实现接口 Quadrable，位于 shape 包中。类的其他部分可根据需要设置，如：定义 Triangle 类的话，可以有表示 3 条边长的数据域。

<3> 定义一个 Client 类，其中的 main 方法中建立一个 ArrayList 和一个数组，调用 ShapeUtils 类中的两个方法计算面积之和。

提交要求：打包为可以执行的JAR文档，其中要包含源程序文件。

## 实验 8 学生成绩分析程序

实验类型：综合性

实验目的：综合运用 Java 语言和面向对象技术开发一个小型软件系统

实验内容：

已经提供的数据：

<1> 班级学生名单：学生的学号和姓名已经分别存储在以学生所在班级命名的文本文件中，每行存储一个学生的学号和姓名，中间用一个空格分隔。例如：2010 级网络工程 1 班.txt 文件中存放该班学生名单。内容为：

```
201030720102 陈纯
201030720103 陈娟娟
201030720104 陈伟健
201030720105 陈伊纯
.....
```

<2> 开设课程保存在名为：course.txt 的文本文件中，每行存放一门课程的名称，如：  
面向对象程序设计  
操作系统  
数据库系统

实现以下功能，所有功能以图形用户界面完成。

<1> 新建课程考试成绩单，功能描述如下：

- 程序界面显示已经开设的课程（从 course.txt 中读取），用户选择本次输入的课程。
- 程序界面通过文件对话框要求用户选择为哪个班输入成绩（即选择相应班的名单文件）。如果该班的成绩已经输入（已经存在对应成绩单文件），则提示无需输入。
- 程序提供界面为该班的每个学生输入考试成绩。
- 输入的成绩单以对象文件格式存储到文件中（不能是文本文件），文件命名为：班级-课程名.dat。例如：2010 级网络工程 1 班-面向对象程序设计.dat。

<2> 打开课程考试成绩单，功能描述如下：

- 程序通过文件对话框要求用户选择打开的成绩单文件。
- 程序打开并读取成绩单文件内容，并显示在界面中。

<3> 修改课程考试成绩单，功能描述如下：

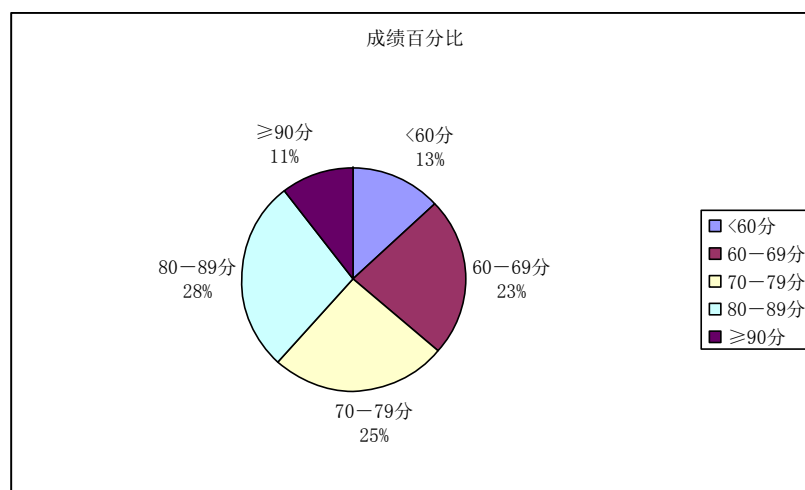
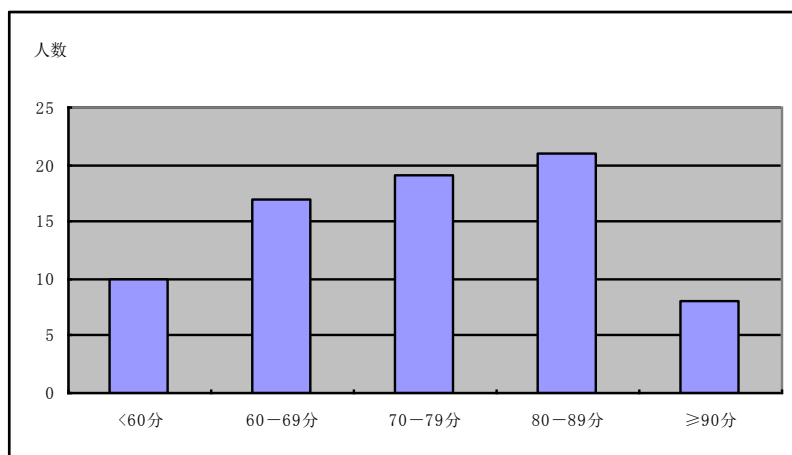
- 完成第<2>功能后，即打开某班某课程的成绩单后。
- 可以选择修改其中某个或某几个考试成绩，并保存。

<4> 课程考试成绩分析，功能描述如下：

- 完成第<2>功能后，即打开某班某课程的成绩单后。
- 点击成绩分析按钮或菜单，显示如下分析内容：
  - ◆ 最高分：XX 分，最低分：XX 分，平均分：XX 分
  - ◆ 不及格（分数<60）：XX 人，占 XX.XX%
  - ◆ 及格（60<=分数<70）：XX 人，占 XX.XX%
  - ◆ 中等（70<=分数<80）：XX 人，占 XX.XX%
  - ◆ 良好（80<=分数<90）：XX 人，占 XX.XX%
  - ◆ 优秀（90<=分数<100）：XX 人，占 XX.XX%

〈5〉 成绩图形分析，功能描述如下：

显示考试成绩分布的饼图和柱形图。如下图：



提交要求：

〈1〉打包为可以执行的JAR文档，其中要包含源程序文件。

〈2〉实验报告，WORD文档，扩展名为.DOC。