**上机实验二**

1. 根据以下的设计要求编写java源代码。

类名: Circle

成员变量为(private): radius(double) //圆的半径

成员方法:

构造方法(没有参数,设置radius的值为10.0)

setRadius(有一个double参数, 将radius的值设为这个新值),

getRadius (没有参数, 返回radius的值)

sameSize(有一个参数, 是另一个Circle对象的引用, 如果两个Circle对象的radius的差小于0.001,则返回true)

在另一个类TestCircle中编写main方法来测试Circle类，创建两个Circle对象c1,c2，调用setRadius方法将c2的半径设置为20.0，调用sameSize方法比较两个对象的尺寸，打印返回结果。

2. 根据以下的设计要求编写java源代码。

类名：Birth

成员变量：year(年)、month(月)、day(日)

构造方法：无参构造方法将成员变量初始化成: year=2000, month=1, day=1(对应日期为2000年1月1日)；带参构造方法对成员变量进行合理初始化，需要对年月日对应值的范围作简单的控制。

成员方法：String toString()，返回对象的出生年月日信息。

要求创建Brith对象，通过构造方法初始化成员变量，输出对象的出生年月日信息及年龄。引入java.util. Calendar类，创建Calendar对象(用Calendar.getInstance()产生Calendar对象)，调用get(Calendar.YEAR)获取当前的年份值。

3. 编写一个复数类Complex验证两个复数1+2\*i和3+4\*i相加产生一个新的复数4+6\*i。复数类Complex必须满足如下要求：

(1) 复数类Complex的属性有：

realPart是int型，代表复数的实数部分

imaginPart是int型，代表复数的虚数部分。

(2) 复数类Complex的方法有：

Complex()构造方法，将复数的实部和虚数都置0

Complex(int r, int i)构造方法，形参r为实部的初值，i为虚部的初值。

Complex complexAdd(Complex a)将当前复数对象与形参复数对象相加，所得的结果仍是一个复数值，将其返回给此方法的调用者。

Complex complexAdd(Complex a, Complex b) 可选

String toString()把当前复数对象的实部、虚部组合成c+d\*i的字符串形式，其中，c和d分别为实部和虚部的数据。

4. 编写一个银行账户类Account，并编写一个main方法，对Account类进行测试，在main方法中要调用Account的所有方法。存取款的步骤：先存入帐号若干元（随机产生），然后取款若干元（随机产生），最后查询帐号信息。

该类的成员变量如下（访问权限均为private）：

|  |  |  |
| --- | --- | --- |
| **变量名** | **含义** | **数据类型** |
| id | 帐号 | String |
| owner | 账户持有人姓名 | String |
| balance | 余额 | double |

该类的成员方法如下（访问权限均为public）：

|  |  |  |  |
| --- | --- | --- | --- |
| **方法名** | **返回类型** | **参数** | **说明** |
| 构造方法 |  | 无 | 构造一个账户实例，将id，owner设为null，balance设为0.00 |
| 构造方法 |  | String id, owner, double amount | 构造一个账户实例，用参数设置成员变量id，owner，balance的值 |
| setID | void | String id | 用参数设置成员变量id的值。 |
| setOwner | void | String owner | 用参数设置成员变量owner的值。 |
| deposit | void | String id, double amount | 将金额amount存入id帐户，如果id帐号为null，则打印信息为“帐号未知！” 否则打印“成功存款×××元,当前余额为×××元！”。 |
| withdraw | void | String id, double amount | 从id帐户支取金额amount，如果帐号为null或者余额小于amount，则打印信息分别为“帐号未知！”和“余额不足！”，否则打印“成功取款×××元，当前余额为×××元！” |
| queryBalance | void | 无 | 打印id，owner，balance。 |

5. 根据类的继承和多态编写程序实现披萨制作。

(1) 定义披萨类：

属性：名称、价格、大小

方法：显示披萨的属性信息

(2) 定义培根披萨和海鲜披萨继承自披萨类

培根披萨的属性：名称、价格、大小、培根克数

方法：显示培根披萨的属性信息

海鲜披萨的属性：名称、价格、大小、配料信息

方法：显示海鲜披萨的属性信息

简化版本：(3) 测试类，分别创建培根披萨和海鲜披萨对象，分别输出培根披萨和海鲜披萨对象的属性信息。

(3) 测试类，主要是根据用户输入的信息制作披萨。

接收用户输入的信息，选择需要制作的披萨。可供选择的披萨有：培根披萨和海鲜披萨。如果选择制作培根披萨，需要用户从键盘输入培根披萨的相关属性，并输出培根披萨的相关属性；如果选择制作海鲜披萨，需要用户从键盘输入海鲜披萨的相关属性，并输出海鲜披萨的相关属性。如果选择其它披萨，则需显示“选择有误，请重新选择”，回到重新选择提示。

(4) 运行效果如下图2-图4所示:

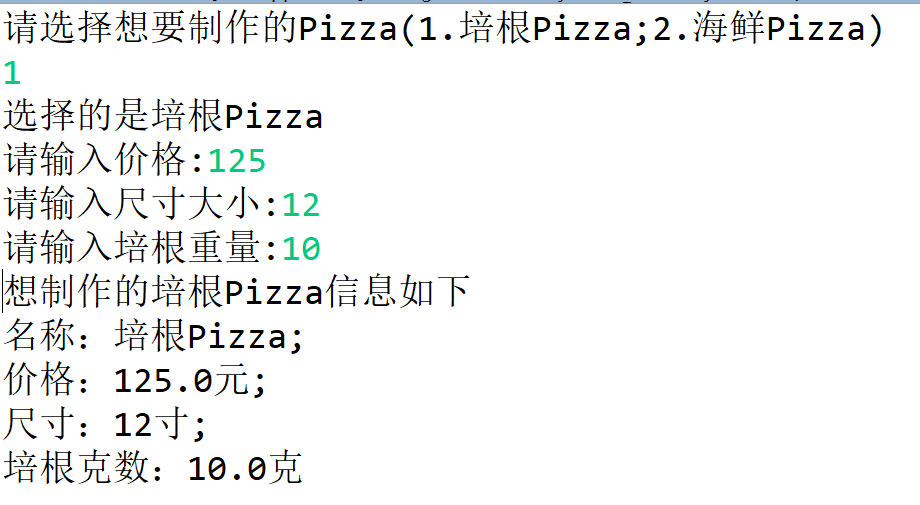


图2

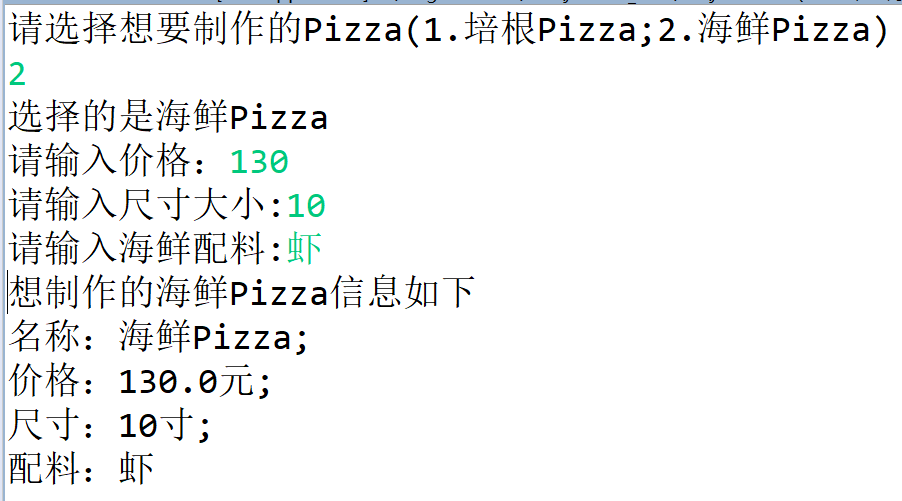


图3

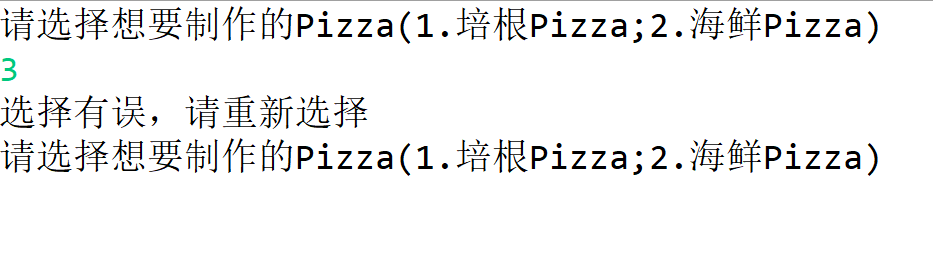


图4