1.面向对象的特征一：封装与隐藏

问题引入：

创建了类的对象以后，我们可以使用“对象.属性”方式调用或设置属性的值。在赋值时，要求考虑到变量的数据类型和储值范围。但是，我们在实际问题，需要额外的给属性赋值时，加入限制条件。这些限制条件不可能在变量声明时做添加，我们只能通过在方法中给变量赋值，同时添加限制条件（setXxx()体现。同时，我们应该禁止直接通过“对象.属性”的方式给属性赋值。
此外，为了能调用此属性，我们再提供获取属性的方法（getXxx()）

2.封装性思想具体的代码体现：

体现一：私有化类的属性，提供公共的get()和 set()去访问和设置

体现二：类中提供私有的方法，表明此方法只能在类内部被调用

体现三：如果此类只希望在本包内被调用，则可声明为缺省状态。

体现四：单例模式

3.java规定的四种权限修饰符---广义上封装性的体现

3.1 权限从小到大顺序为：private < 缺省 < protected < public。

3.2 具体的修饰范围：

修饰符	类内部	同一个包	不同包的子类	任何地方
private	Yes			
(缺省)	Yes	Yes		
protected	Yes	Yes	Yes	
public	Yes	Yes	Yes	Yes

说明：任何地方：本project内有效

3.3 权限修饰符可用来修饰的结构说明：

修饰相应的结构，体现这个结构被调用时，可见性的大小。

3.2 具体的修饰范围：

修饰符	类内部	同一个包	不同包的子类	任何地方
private	Yes			
(缺省)	Yes	Yes		
protected	Yes	Yes	Yes	
public	Yes	Yes	Yes	Yes

说明：任何地方：本project内有效

3.3 权限修饰符可用来修饰的结构说明：

1. 4种权限可以修饰类内部的成员：属性、方法、构造器、内部类
2. 类本身可以使用2种权限修饰： 缺省 、public

说明

标题：类的结构：构造器

1.构造器（或构造方法）：Constructor

构造器的作用：

创建类的对象；初始化对象的属性

2.使用说明：

```
* ① 如果我们没显式的提供类的构造器的话，则系统会默认给一个类提供默认的构造器：无参数的。
* ② 我们如果显式的声明类的构造器的话，格式为：权限修饰符 类名(形参列表){}
* ③ 同一个类中的多个构造器之间构成重载。
* ④ 如果我们显式的提供了类的构造器，则系统不再提供默认的空参的构造器
* ⑤ java的类中一定存在构造器.
```

4.9 关键字—this

this是什么？

- 在java中，this关键字比较难理解，它的作用和其词义很接近。
  - 它在方法内部使用，即这个方法所属对象的引用；
  - 它在构造器内部使用，表示该构造器正在初始化的对象。
- this表示当前对象，可以调用类的属性、方法和构造器
- 什么时候使用this关键字呢？
  - 当在方法内需要用到调用该方法的对象时，就用this。

  具体的：我们可以用this来区分局部变量和属性。比如：

  this.name = name;

```
2
3  this关键字的使用
4  1. 理解为：当前对象 或 当前正在创建的对象
5
6  2. 使用范围：this可以使用在方法或构造器的内部。
7
8  3. 可以调用的结构：属性、方法；构造器
9
10 4. 在方法中使用this:
11    我们可以在方法内，调用当前类的属性或方法，属性或方法前可以使用"this."的方式，表明调用的是当前对象的属性或方法。
12    通常情况下，我们都习惯省略此this关键字。
13    但是，如果方法的形参和类的属性名相同了，则必须使用"this.变量"的方式，表明我们调用的变量是当前对象的属性，而非形参。
14
15 5. 在构造器中使用this:
16    我们可以在构造器内，调用当前类的属性或方法，属性或方法前可以使用"this."的方式，表明调用的是当前对象的属性或方法。
17    通常情况下，我们都习惯省略此this关键字。
18    但是，如果方法的形参和类的属性名相同了，则必须使用"this.变量"的方式，表明我们调用的变量是当前对象的属性，而非形参。
19
20
```

```
21  * 6. this来调用构造器:
22  * ① 我们可以在构造器中使用“this(形参列表)”方式，调用当前类中的指定构造器
23  * ② 构造器中不可以使用“this(形参列表)”方式调用本身的构造器
24  * ③ 如果类中声明了n个构造器，则最多可以有n - 1个构造器中使用了"this(形参列表)"方式调用其他构造器
25  * ④ 规定：要求将"this(形参列表)"声明在构造器的首行
26  * ⑤ 一个构造器中最多声明一个"this(形参列表)"
27  */
```

练习7

添加必要的构造器，综合应用构造器的重载，this关键字。

Boy
-name:String -age:int
+setName(i: String) +getName(): String +setAge(i: int) +getAge(): int +marry(girl:Girl) +shout():void

Girl
-name:String -age:int
+setName(i: String) +getName(): String +marry(boy:Boy) +compare(girl:Girl)

1、写一个名为 Account 的类模拟账户。该类的属性和方法如下图所示。该类包括的属性：账号 id，余额 balance，年利率 annualInterestRate；包含的方法：访问器方法（getter 和 setter 方法），取款方法 withdraw()，存款方法 deposit()。

Account
private int id private double balance private double annualInterestRate
public Account (int id, double balance, double annualInterestRate)

public int getId()
public double getBalance()
public double getAnnualInterestRate()
public void setId(int id)
public void setBalance(double balance)
public void setAnnualInterestRate(double annualInterestRate)
public void withdraw (double amount)//取钱
public void deposit (double amount)//存钱

提示：在提款方法 withdraw 中，需要判断用户余额是否能够满足提款数额的要求，如果不能，应给出提示。

JDK中主要的包介绍

1. java.lang----包含一些Java语言的核心类，如String、Math、Integer、System和Thread，提供常用功能
2. java.net----包含执行与网络相关的操作的类和接口。
3. java.io ----包含能提供多种输入/输出功能的类。
4. java.util----包含一些实用工具类，如定义系统特性、接口的集合框架类、使用与日期日历相关的函数。
5. java.text----包含了一些java格式化相关的类
6. java.sql----包含了java进行JDBC数据库编程的相关类/接口
7. java.awt----包含了构成抽象窗口工具集（abstract window toolkits）的多个类，这些类被用来构建和管理应用程序的图形用户界面(GUI)。 B/S C/S Client

```
/*
 * 一、package关键字的使用
 * 1.package : 包
 * 2.作用：使用包的概念来区别不同类的功能。换句话说，将相同结构或类型的类等放在相同的包中，便于管理。
 * 3. package 声明在源文件的首行
 * 4. 命名的规范：xxxyyy.zzz.kkk
 * 5. 每"."一次就相当于一层文件目录
 *
 *
 * 二、import关键字的使用
 * 1. import:导入
 * 2. 我们可以在源文件中使用import显式的导入指定包下的类或接口
 * 3. 声明位置：通常声明在package包声明和类文件之间
 * 4. 如果使用的类或接口是当前包下的，则可以省略import的操作
 * 5. 如果使用的类或接口是java.lang包下的，则也可以省略import的操作
 * 6. 如果需要导入多个结构，则并列声明出即可。
 * 7. 可以使用“包名1.*”的方式，导入包名1下的所有结构。
 * 8. 如果类中出现了不同包下的同名的类，则至少有一个需要使用全类名的方式进行调用
 *
 * 9. import static：导入指定包下的指定类或接口中的静态结构
 * 10.已经导入了a包下的所有结构，则如果需要导入a包的子包的结构的话，仍需要import操作
 */
```