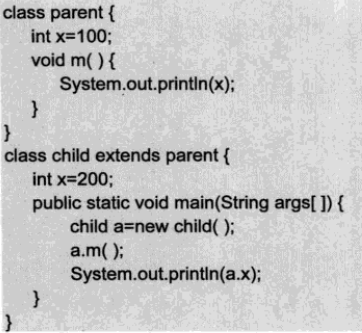
**实验三（2） 继承和访问控制**

1. 实验目的
2. 理解类的继承，掌握变量隐藏、方法覆盖的概念
3. 理解多态，掌握方法的匹配调用原则
4. 理解this, super的含义及使用
5. 掌握访问控制符的使用
6. 实验内容
   1. **类的继承**

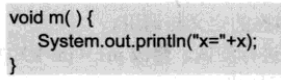
通过类的继承，祖先类的所有成员均将成为子类拥有的“财富”。但能否通过子类对象直接访问这些成员取决于访问权限的限制。Object类是所有类的祖先。

1.继承关系中的覆盖与重载问题



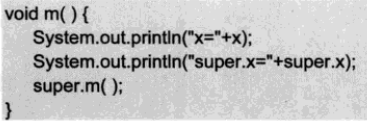
调试程序，分析结果，理解方法属性的继承与父类属性隐藏的概念。

（2）在子类中增加一个方法，和Parent类一样，为了显示差别，在输出上加处理。



重新编译运行程序，观察结果变化，理解方法的覆盖关系

（3）在子类方法中，加入super引用访问父类方法和属性的代码



观察结果。理解super访问的特点

（4）站在父类引用看成员



观察结果变化，总结用父类引用子类对象

（5）将以上属性和方法改为静态属性和方法，重新进行测试，总结规律。

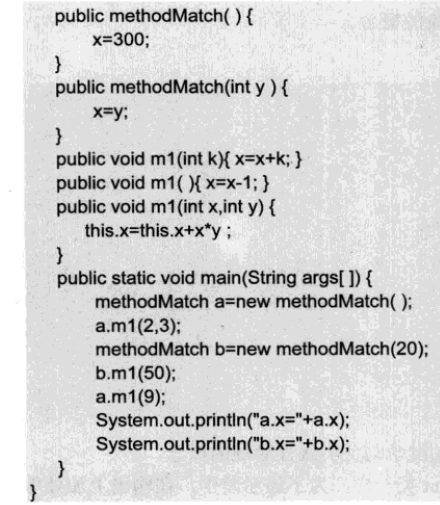
* 1. **多态的表现**

多态的两种表现形式：（1）同一类中的方法多态（方法重载）（2）子类对父类方法的重新定义（方法覆盖）。

方法调用的匹配原则：首先查找是否有参数一致的方法，也就是精确匹配；如果没有，再检查实参是否能自动转换为形参类型，能转换也可以匹配调用，这种匹配称为转换匹配。

方法的参数多态

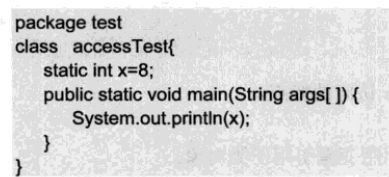




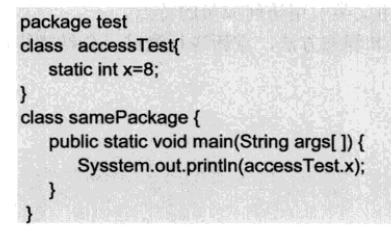
1. 调试程序，理解构造方法的参数多态与方法的参数多态的定义与调用。
2. 将main方法中的b.m1(50)改为b.m1(50.2)进行测试，修改m1(int k)方法的参数为double类型，重新调试程序，总结参数转换匹配规律。
   1. **访问控制符**

分别对下面程序中x的访问权限进行修改，测试访问许可，修改包括private、protected、public以及无访问控制符4种情形。

（1）同一类情形（假设文件accessTest.java）

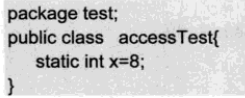


(2)同一包情形（文件samePackage.java）

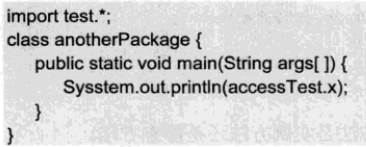


(3)不同包的其他类情形

文件1：asscessTest.java

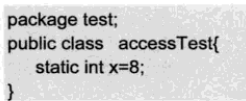


文件2：anotherPackage.java

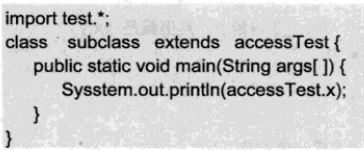


(4)不同包的子类情形

文件1：accessTest.java



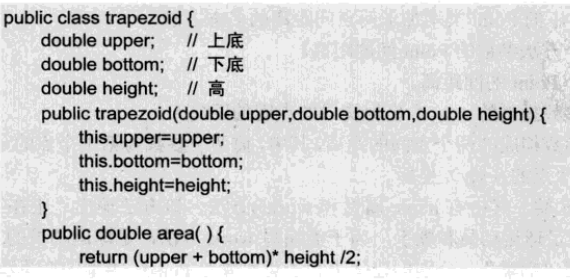
文件2：subclass.java

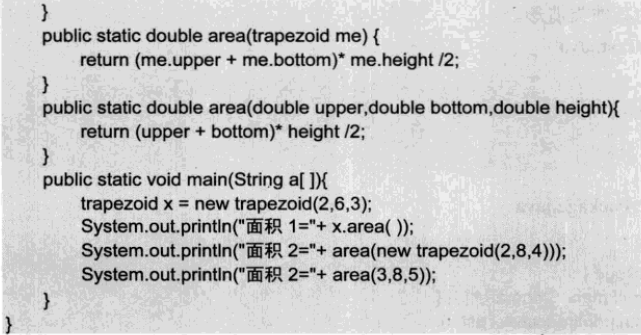


对照课堂上给的访问控制权限开放表，是否结论一致？

* 1. 综合样例

看看下面的梯形对象，体会一下求梯形面积的参数多态方法。注意实例方法还是静态方法。





* 1. **自测题**

1. 按以下描述和要求设计一个课程类Course，它封装了：

四个成员变量：

courseID //课程代号

couseName //课程名称

credit //课程学分

teacher //任课老师

五个重载的构造器：

Course() //默认的构造器

Course(String courseID) //单参数的构造器

Course(String courseID, String courseName) //两个参数的构造器

Course(String courseID, String courseName, int credit) //三个参数的构造器

Course(String courseID, String courseName, int credit, String teacher) //四个参数的构造器

要求：

（1）Course(String courseID, String courseName, int credit)构造器要调用构造器Course(String courseID, String courseName)；//提示用this

（2）构造器Course(String courseID, String courseName, int credit)要调用Course(String courseID, String courseName, int credit, String teacher)。//与上题不同，参数少怎么去调用参数多的？提示，可以提供默认值

2. **继承和派生**

（1）设计一个Employee类，包含姓名，年龄，部门，参加工作时间（日期型，精确到日）和基本薪水（basicSalary）这5个属性，包含一个计算总薪水（Salary）的方法(comSalary)，该方法根据当前时间和其参加工作时间的间隔年数（dyear），用Salary= basicSalary+dyear\*20.

（2）设计一个Manager类，该类从Employee类派生，具有分红（bonus）属性，重写计算总薪水（Salary）的方法(comSalary)，计算公式为Salary= basicSalary+dyear\*20+bonus

（3）设计一个Contractor类，该类从Employee类派生，具有工作结束时间属性，重写计算总薪水（Salary）的方法(comSalary)，计算公式为Salary= basicSalary+（工作结束时间-参加工作时间）\*15。

（4）新建一个DiverseClass，为其设计一个函数，传入任意一个Employee、Manager、Contractor对象，都能正确调用各自的计算总薪水的函数。