## # 韩顺平javase基础教程

- javase是javaee和javame的基础，必须重视

- java一般是操作Oracle数据库

- java多线程实例：坦克大战

分别可以用HTML5、js、java来实现

> 学习必须认真，多思考，多做笔记

\* 学习软件编程是在琢磨别人怎么做，而不是我认为应该怎么做的过程

-----------------------------------------

什么是内部类，什么是多线程，io ？？？

- java主要是用来做后台的，并不擅长做界面

- 因为java和Oracle都是甲骨文公司的，所以，java主要用的是Oracle数据库

- 网络编程就是socket编程，所有的编程语言（python,php）都可以进行网络编程

-----------------------------------------

多线程的实现方式：

1. new thread()

2. 实现runnable接口

-----------------------------------------

- 人和计算机交流就要用编程语言，编程语言通过转化成二进制机器语言操作计算机硬件

- NetBeans是sun公司的，jcreate早已退出历史舞台

java语言的特点：简单、面向对象、跨平台、高性能的

-----------------------------------------

开发环境介绍：jdk：java开发工具包

jdk包含：

1. jre：java运行环境

2. java的工具，比如java的编译器javac.exe java的解释执行器 java.exe

3. java的类库(4000多个类)常用的就150多个

> jre只是运行环境，jdk是开发工具

只要有jre即可运行编译好的java程序

> 注意：

1. 在java中单引号只能包含一个字符（一个字母或者汉字）字符串就必须用双引号！！！

2. 局部变量不能重复！！！

错误：

D:\javaproject\come\test>java Demo1

错误: 找不到或无法加载主类 Demo1

出现这个错误的原因有：

1.没有设置好 classpath引起的，

2.类名输入错了

3.文件名和类名不一致

> 可用set path 查看当前环境变量的设置

只有以下文件类型才能在path中设置

PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC

在控制台设置临时classpath

set classpath=d:\javaproject

set classpath=d:\javaproject;

注意这两句是有区别的，第二句还会去当前目录去找！！！

告诉虚拟机class文件的路径，这样虚拟机就会到指定目录去找类文件，而不用进入到该目录

set classpath=. 这个点是一个变量，你进入到哪个目录，他就是哪个目录

不建议在系统的环境变量中设置classpath，如果设置不当，就算进入了指定目录也是找不到类文件的，jvm只会去系统环境变量中设置的路径去找，根本不会到你进入的当前目录去找！！！

%classpath%表示系统原来的classpath

编译出来的是类文件，有多少个类就会编译出多少个类文件！！！

有个问题：用eclipse写的java在控制台编译过去但是运行不了？

第一个java程序

注意：文件名 Hello.java 类名必须与文件名一致

一个java文件只能有一个public类

public class Hello

{

public static void main(String args[])

{

System.out.println('hello world');

}

}

先编译，后运行

先javac -encoding utf-8 Hello.java //编译的时候是全路径

再java hello //运行时，只需要一个类名

问题：提示找不到或无法加载主类？

这是由于没有设置classpath造成的，classpath就是指定生成的class文件的路径，如果没有配置环境变量，就要到指定目录运行才能成功！！！

一般的解决方法是：在.java 文件目录下直接运行

配置classpath环境变量：

classpath这个参数，设为 .;%java\_home%\lib\dt.jar;%java\_home%\lib\tools.jar

看清楚有两个分号，第一个前面只有一个点，表示的是当前目录，这个很重要，后面两个是两个第三方jar包，这个参数的值不包括标准类库和扩展类库的jar包，因为标准类库和扩展类库是默认加载的不需要设置，要设置额外的第三方类

-----------------------------------------

### 二进制相关知识

00000000 最高位是符号位

0表示整数

1表示负数

1B=8b 100Mb=100/8MB=12.5

正数：0111 1111 表示127

负数：1111 1111 表示-128

可以这样理解:

对于0

0000 0000 +0

1000 0000 -0

-0是没有必要的，所以印度人就把-0看成是-128，这正好与补码相吻合

对于int long 都可以这样推出他们的具体范围

以一个字节举例：

补码：对于正数，原码，反码，补码 都是自身

对于负数：原码1111 1100

反码：符号位不变，其余为取反

1000 0011

补码:在反码基础上末位+1

1000 0100

所以表示的十进制负数为：-4

原理上是这样推，实际上可以这样计算：除了符号位，(/后面各位之和/)-128

也可以：-(128-/后面之和的绝对值/)

也就是 -128+后面绝对值 所以最小为-128

2^7=128 != 01111111=127 所以此二进制数对应的十进制应该为2^7-1=127

如果忽略符号位：也就是全部位都用来表示范围

11111111 表示十进制数为：2^8-1=255

一定要弄清楚，二进制数可以表示的范围和可以表示的组合，这是两个不同的概念！！！

如果单说组合，就忽略符号位，由于每一位有两种可能，所以 2\*2\*2...\*2,也就是有n位，就有2^n种组合

组合数 总比可以表示的无符号最大正数大1，因为0也是一种组合！！！

如果是四个字节又是怎么推出它的范围呢？

思考过程：四个字节，共有4\*8=32位，除了符号位，有31位

所以可以表示十进制正数为：2^31-1=2147483647

可以表示的最大负数为：-2147483648

最小负数的绝对值总比最大正数的大1，因为+0给了正数部分，负数部分就必须多出以为来代替-0，所以绝对值大1！！！

所以最终版本公式：-2^(位数-1)————2^(位数-1)-1

解释：正数部分和负数部分占同样的一半，而正数占多了一个0的位置，所以要 -1

最好以数轴来形象理解！！！

byte表示一个字节，8位

short：两个字节 -32768-32767

int：四个字节

long：八个字节 -9223372036854775808--9223372036854775807

如果是一个海量数，超过java所能存放的范围呢？

浮点类型：

float：

double：

String不是简单的数据类型了，是 类

字符类型char可以表示单个字符

char是两个字节，可以存放一个汉字

如果是单纯的字符相加，则会按照ascii相加，返回数字，然后可以根据前面的 类型限制，自动转为相应类型

结论：在java中，对插入进行运算时，直接当做ascii码对应的整数对待

多个字符，则要用String类来表示

java是强类型，在定义变量时，就要限定类型

这一点与js后者PHP不一样

-----------------------------------------

基本数据类型的转换：

1.自动转换

int a = 1.2;//提示损失精度

也就是数据类型不能自动从高精度转成低精度

低精度是可以自动转成高精度的

所谓的精度就相当于不同规格的容器：高精度对应大容器；低精度对应小容器

小容器是装不下大容器里面的物体的

float a = 3.2;//3.2默认double类型，不能往float转

float a = 3.2f;//正确，f明确指定这个数为单精度

2.强制转换

先转换再赋值，比如

int a = (int) 1.2;//1

int b = (int) 1.9;//1

说明，java并不会四舍五入

int a =1;

int b = 1.2;

a = b;

以上也是不可以的，java在编译时会检测出来

计算过程中的转换

int a = 3;

int b = a +3.4;//这里已经变成了double，也就是结果往高精度转换，所以这里结果为double类型

System.out.println(b);

-----------------------------------------

运算符：

取模也就是取余数 %

1、两个数能整除 0

2、两个数不能被整除 余数

3、如果 被除数<除数 结果为 被除数

比如： 4%5=4 也就是商为0，余数为4

java中同样有自加、自减

也就是先自身加还是先返回的意思

system.in 标准输入，也就是我们的键盘

接受键盘的输入流：

InputStreamReader isr = new InputStreamReader(system.in);

BufferedReader br = new BufferedReader(isr);

String str = br.readLine();

将字符串转换成float

float num = Float.parseFloat(str);

如果控制台提示“找不到符号，很可能是没有引包” 所以要在开头：import javax.\*;

包就是用来管理类的，包里面有很多的类，如果要使用某一个类，必须要把这个类所属的包引进来！！！

如果编译出现乱码，可以指定字符集，

javac -encoding utf-8 Hello.java

编译也是严格区分大小写的！！！

关系运算符 < > 只能用于比较数，不能用于比较字符串

1）基本数据类型：byte、short、int、long、float、double、char、boolean

2）引用数据类型: 数组、类、接口。

级别从低到高为：byte,char,short(这三个平级)-->int-->float-->long-->double

自动类型转换：从低级别到高级别，系统自动转的；

小范围的赋给大范围的 可以直接赋值

大范围的赋给小范围的 可能损失精度 需要强转（有可能出现数据的溢出）

1.注意：若数值过大需加L或l表明是long类型

2.小数字面常量默认是double类型

3.注意：float类型存小数时需要加f或F

-----------------------------------------

#### 逻辑运算符：

真 假

与&&

或||

非 !

#### 转义字符

\b 退格

\t 制表

\n 换行

\r 回车

\” 双引号

\’ 单引号

\\ 反斜杠

> %:任何整数模2不是0就是1 也就是奇偶数

#### 位运算符

&: 只有两边都为true结果是true。否则就是false。

|:只要两边都为false结果是false，否则就是true

^:异或：和或有点不一样。两边结果一样，就为false。两边结果不一样，就为true.

& 和 &&区别：

& ：无论左边结果是什么，右边都参与运算。

&&:短路与，如果左边为false，那么右边不参数与运算。

| 和|| 区别

|：两边都运算。

||：短路或，如果左边为true，那么右边不参与运算。

-----------------------------------------

流程控制中的多分支，与PHP，js类似，java中也是有switch语句的

具体语法如下：

switch(表达式){

case 值1：

代码块;

break;

//break 表示退出这个switch，所以每个case部分都必须要有break

case 值2：

//注意这里是值而不是表达式或者关系运算表示的一个范围

代码块;

break;

default:

代码块;

break;//注意：最后一个default是可有可无的

}

对于switch多分支，需要注意的是：表达式的值类型必须要与值类型一致！！！

表达式的值可以是 byte int short char enum枚举类型

但是不可以是float long

在两个case代码块之间可以书写任何代码块

细节：

1. break是可以省略的，如果省略了就一直执行到遇到break为止；

2. default可以写在switch结构中的任意位置；如果将default语句放在了第一行，则不管expression与case中的value是否匹配，程序会从default开始执行直到第一个break出现。

3. 当判断数据范围，获取判断运算结果boolean类型时，需要使用if。

4. 当某些语句需要执行很多次时，就用循环结构。

-----------------------------------------

while和for可以进行互换。

区别在于：

如果需要定义变量控制循环次数。建议使用for。因为for循环完毕，变量在内存中释放。

break:作用于switch ，和循环语句，用于跳出，或者称为结束。

break语句单独存在时，下面不要定义其他语句，因为执行不到，编译会失败。当循环嵌套时，break只跳出当前所在循环。要跳出嵌套中的外部循环，只要给循环起名字即可，这个名字称之为标号。

continue:只作用于循环结构，继续循环用的。

作用：结束本次循环，继续下次循环。该语句单独存在时，下面不可以定义语句，执行不到。

return:退出方法或结束程序。

-----------------------------------------

函数的作用：

1）、用于定义功能。

2）、用于封装代码提高代码的复用性。

注意：函数中只能调用函数，不能定义函数。

-----------------------------------------

变量的 定义、声明 是不能重复的，即是不能

char a; int a;

循环语句，就是让你的代码可以循环执行

1 2 4

for(int i=0 ;i<10;i++){

//3

System.out.println("我是刘德华");

}

如上循环的执行流程：1->2->3->4

也就是，先执行第一步，初始化一个i，第二步，判断表达式是否为真，若为真，执行4循环体，否则退出循环，执行完循环体后，执行第三条语句，自增，然后再回到第二条语句进行判断...

第一条语句只执行一次！！！

while循环

while(条件表达式)

{

//只要条件为真就一直执行

代码部分+跳出循环语句

}

do

{

代码块+跳出循环语句

//至少执行一遍，这种用法比较少

}while(条件表达式);//如果为真，继续执行循环体

打印金字塔：

先打一半，

思路：死去活来发，先固定行找出空格与 \* 号与当前行i之间的对应关系

然后在一个循环中打印空格，另一个循环中打印 \*

public class Demo2

{

public static void main(String[] args) {

int lay = 4;

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

for (int j=1;j<=lay-i ;j++ ) {

System.out.print(" ");

}

for (int k=1;k<=2\*i-1 ;k++ ) {

System.out.print("\*");

}

System.out.println();

}

}

}

如果是要打印一个镂空的金字塔呢？

思路：在循环打印内部 \* 时，做判断

顶层和底层原样输出

if(i==1 || i==lay){星号}

else{

//判断第一个和最后一个

if(j==1 || j==2 \* i - 1){星号}else{空格}

}

也就是在用for循环打印\*号是，判断过滤打印！！！

-----------------------------------------

### 面向对象概述

- 面向过程思想：强调的是功能和行为

- 面向对象对象:将功能封装进对象，强调具备功能的对象

面向对象的三个基本特征：封装 继承 多态

什么是类：类就是对现实生活中事物的描述。描述事物其实就是在描述事物的属性行为

什么是对象：对象就是这类事物，实实在在存在个体

为什么要定义类：根据需求，以后的开发其实就是在找对象，没有就要创建一个对象

对象必须具备：属性和方法

面向对象的特点：找对象，建立对象，使用对象，维护对象的关系

### 面向对象内存分析

Cat c=new Cat();

#### 面向对象的一些知识点

类中的属性被称为成员变量(每个对象的属性都不同)

类中的方法被称为成员方法(每个对象共享类中的方法)

方法的种类：

1.无参无返回 public void get(){}

2.无参有返回 public int get(){}

3.有参无返回 public void get(string A){}

4.有参有返回 public String get(int a){}

根据需求来定义属性和方法

关键字：

return 返回语句，被用于返回

this 用来指当前对象

private 隐藏类中方法或属性对外界的可见性,常用于封装

toString每个类的一个默认方法,用于把类以字符串显示

类是抽象的，概念的，代表一类事物

对象是具体的，实际的

类是对象的模板，对象是类的一个实体

定义一个完整的类

package 包名;

class 类名 extends 父类 implements 接口名{

成员变量;

构造方法;

成员方法;

}

创建对象是会占用内存的，靠gc来回收内存

当java虚拟机发现一个对象没有一个引用指向他时，就认为这个对象是垃圾

销毁对象的方法：a = null;gc会在适当的时候回收内存

一般是当jvm发现内存不够了，就进行回收

在c中，在哪里开辟了内存，就在哪里回收

java还是有可能出现内存泄漏的情况的，但是几率比较小

对象是按照引用传值的，一改全改

成员方法：

java中每个函数都有一个作用域，规定成员变量只能在该作用域内生效

java是支持重载的，java认为同名不同参函数，是可以共存的，根据调用的时候是否传入参数，调用相应的方法

java函数执行流程图：

main->自定义成员函数

只要有新函数被调用了，jvm马上开辟一个栈，完全独立main函数，当函数运行完之后，重新回到主函数 ，main函数执行完之后就退出程序

两个同名函数如果只有返回值类型不一样，是不能编译成功的

注意：return的类型要和函数的类型一致

方法的声明是没有函数体的，没有 {}

也就是 public int Test(int a);

return 表达式;把return的值返回给主调函数

实参的类型必须要与形参定义的参数类型一致！！！

一定要注意2.2这样写在java中是double类型，要表示浮点数要这样写2.2f

注意主类和辅助类的区别：

main方法和其他普通方法的区别：

命令行使用技巧：打开快速编辑，这样就允许鼠标拖动进行选中，右键就是复制选中的内容！！！

java中的所有变量都必须给他指定一个类型

出现这个错误：

D:\javaproject>javac -encoding utf-8 Demo3.java

Demo3.java:4: 错误: 无法从静态上下文中引用非静态 方法 jisuan1(int)

解释：也就是说用static修饰的方法，只能调用用static修饰的方法！！！

静态方法内部只能调用静态方法！！！

在java中可以直接通过 类名.静态方法名 来直接调用类中的静态方法，而无需new对象！！！

错误：

D:\javaproject>javac -encoding utf-8 Demo3.java

Demo3.java:30: 错误: 未报告的异常错误IOException; 必须对其进行捕获或声明以便抛出

解决：如果方法有IO操作，有两种解决方法，第一种就是在这个方法的声明后面加上 throw IOException;

比如：public static void main(String[] args) throws IOException

第二种解决方法：使用try{包裹有可能出现异常的代码}catch(IOException){捕获到异常，然后要执行的操作}

错误：

Demo4.java:8: 错误: jisuan1(int)可以在Calculator中访问private

解释：calculator中的jisuan1()方法是私有的，只能在这类的内部访问

如果实在要在类外访问就只能在这个类里面再定义一个public方法，然后在这个public方法中调用本类的私有方法

学习要求：理解好每一个知识点！！！不能有半点马虎！！！

-----------------------------------------

打印乘法表：

//功能：写一个函数完成乘法表

public class Demo3{

public static void main(String[] args) {

//注意：这个是类的主要方法，所有的业务逻辑都在此方法内执行，这里一般不会出现很多代码

jisuan1(3);

}

public static void jisuan1(int n){

//控制有多少行

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

//控制每一行有多少列

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

System.out.print(j+"\*"+i+"="+j\*i+" ");

}

//打印完一行的所有列之后，打印一个换行

System.out.println();

}

}

}

-----------------------------------------

//从命令行接收一个参数，打印乘法表

import java.io.\*;

public class Demo3{

public static void main(String[] args) throws IOException {

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("请输入一个数，用来计算乘法表");

int n = Integer.parseInt(br.readLine());//readLine()读取一行,这是具体的方法，只能在对象中调用

jisuan1(n);

}

public static void jisuan1(int n){

//控制有多少行

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

//控制每一行有多少列

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

System.out.print(j+"\*"+i+"="+j\*i+" ");

}

//打印完一行的所有列之后，打印一个换行

System.out.println();

}

}

}

-----------------------------------------

//功能:在类中封装一个方法，完成输出乘法表功能

public class Demo4{

public static void main(String[] args) {

Calculator cal = new Calculator();

cal.jisuan1(6);//通过对象来调用具体的方法

}

}

//定义一个新的工具类或者叫辅助类

class Calculator{

//通过修改这里了的public protected private可以得出权限范围

protected void jisuan1(int n){

//这里如果是static呢？在main中直接Calculator.jisuan1(5);//直接调用类的静态方法

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

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

System.out.print(j+"\*"+i+"="+j\*i+" ");

}

System.out.println();

}

}

}

-----------------------------------------

构造方法：在对象刚被创建时，就开始执行相当于PHP中的 public function \_\_construct(){}

在java中，以类名作为构造方法名 比如一个Person类 构造方法为：public Person(参数){}

注意：java中的构造方法是不能有返回值的

构造方法有两种，一种带参数的构造方法，另一种不带参数构造方法

如果没有定义构造函数，jvm自动调用默认的构造函数来创建对象，但是如果自定义了构造函数，那么在new对象时，必须传入参数！！！

java中构造方法也是方法，也就是说可以重载，即是，可以有多个构造方法，从而可以实现不同的new法，得到不同的对象，如果不希望默认的构造方法被覆盖，则要写出来 public Person(){};

-----------------------------------------

java是完全面向对象的，所有的操作都在类里面进行，类就是一个容器的抽象，里面有很多属性和方法

在一个java文件中各个类是相互独立的

class Test1{}

class Test2{}

class Demo5 {

public static void main(String[] args) {

System.out.println(new Person("朱明浩").name);

}

}

-----------------------------------------

class Person {

String name;

//注意：构造方法不能有返回值，要严格等于类名，必须大写！！！

public Person(String name){

this.name = name;

}

}

-----------------------------------------

问题：

D:\javaproject>javac -encoding utf-8 Demo5.java

Demo5.java:7: 错误: 需要class, interface或enum

public void sayHello(){}

此错误说明，java中不能直接在类外部写方法

Java是完全面向对象的编程语言，而不是面向方法，它是通过类来实现各种操作。单纯的独立于类之外的一个方法，无法实现其作用，也没有意义

-----------------------------------------

来自百度的解答：

main函数是一个程序的入口，也是出口，main函数其实就是类里的一个方法，方法只能在类里声明了，所以main函数必须声明在一个类里，一个程序里只有一个main函数，因为类与类之间是相与调用的。一个程序从开始运行就会先找main函数，然后再一步一步的执行，如果有多个main那么程序就会报错，因为他不知道该从那个门口（main函数）进，然而类为什么可以有多个呢，你还记的JAVA的三大基本特征么，即封装，继承，多态。而封装就是我们所说的类，他是把所有相同的属性，方法归纳到一起，即组成一个类。

主函数的一般写法如下：

　　public static void main(String[] args){…}

　　下面分别解释这些关键字的作用：

　　（1）public关键字，这个好理解，声明主函数为public就是告诉其他的类可以访问这个函数。

　　（2）static关键字，告知编译器main函数是一个静态函数。也就是说main函数中的代码是存储在静态存储区的，即当定义了类以后这段代码就已经存在了。

如果main()方法没有使用static修饰符，那么编译不会出错，但是如果你试图执行该程序将会报错，提示main()方法不存在。因为包含main()的类并没有实例化（即没有这个类的对象），所以其main()方法也不会存。而使用static修饰符则表示该方法是静态的，不需要实例化即可使用。

　　（3）void关键字表明main()的返回值是无类型。

-----------------------------------------

注意：我们自定义的类也可以算是一种类型，也可以在类内部做声明属性时用 比如 Dog dog;这个是引用类型，在类内部声明一个dog属性，它的类型是Dog

如果一个属性是引用类型的，如果被修改了，则这个类的所有对象的这个引用类型属性都被修改

例子：

有个Person类，有两个对象p1和p2

p1修改dog的name属性：p1.dog.name="旺财";

此时，p2查看自己的dog.name : System.outprintln(p2.dog.name);

发现，名字已经变成旺财了！！！

this：java虚拟机给每一个对象分配一个this ，代表本对象

错误：

D:\javaproject>java Demo6

Exception in thread "main" java.lang.NullPointerException

at Demo6.main(Demo6.java:4)

这是一个空指针异常，也就是这个指针没有指向对象，应该创建对象

-----------------------------------------

class Demo6 {

public static void main(String[] args) {

Dog dog = new Dog();//要用到dog,所以要实例化

dog.showinfo();//创建时的名字

Person p1 = new Person("朱明浩",dog);

p1.dog.name = "旺财";//通过对比发现，p1修改dog的name会直接影响p2的dog的name属性

dog.showinfo();//修改后的名字

Person p2 = new Person("陈华",dog);

System.out.println(p2.dog.name);//其他人的dog也变了

}

}

class Person {

Dog dog;//引用类型属性，指向内存中的一个Dog实例化对象

String name;

public Person(String name,Dog dog){

this.name = name;

this.dog = dog;

}

}

class Dog{

String name = "小白";

public void showinfo(){

System.out.println(this.name);

}

}

-----------------------------------------

注意类名不能有 - 但可以是 \_

用static定义的属性，是静态变量，是类级别的变量（可以通过类名访问），如果这个类的任意一个对象修改了这个static定义的属性，那么其他对象的此属性也将改变

static 属性 放在单独的一个静态区

静态代码块

static{}只会执行一次，只是在类声明是调用

static定义的属性，方法，静态代码块都是在类定义时就被创建起来，而与对象是否创建无关！！！

//测试静态代码块的执行顺序

class Demo7 {

static {

System.out.println("静态代码块");

}

public Demo7(){

System.out.println("构造方法");

}

public static void main(String[] args) {

Demo7 d = new Demo7();//在类内部居然可以new自身！！！

}

}

先打印静态代码块，在打印构造方法，没有创建对象时，也会打印静态代码块

-----------------------------------------

使用static定义属性、方法的好处，是类级别的，所有对象共享，就只有一个，大大节省内存开销

如果是普通的，非静态的，每创建一个对象，都会为这个对象开辟新的内存空间，极大浪费资源！！！

java中的规则：静态属性用静态方法去访问，这样理解起来更符合逻辑 虽然也可以用普通方法访问

静态方法只能访问静态属性！！！

-----------------------------------------

#### 封装、继承、多态

包的作用：

1.区分相同名字的类

2.可以很好的管理类

3.控制访问权限

语法：package 包名 ;

包必须放在类定义的最前面个！！！

所谓的jar包就是别人写好的一些类，打包之后，你引入了别人的jar包，你就可以用别人写好的类，大大节省时间！！！

我们常用的包都打包在 rt.jar 中

包名也就是这个类所在的路径，如果直接new对象时报错可以考虑，引包

引入包的目的是使用这个包下面的类

-----------------------------------------

按照权限从低到高有：

private -> default -> protected ->public

权限：同类 同包 子类 不同包

private：同类

default：同类+同包

protected：同类+同包+子类

public：同类+同包+子类+不同包

在不同包，只有public能被访问

所谓权限是指能否被外部访问的权限

-----------------------------------------

#### 继承：

为什么要使用继承呢？

- 提高代码的复用性 子类拥有父类的全部可见方法和属性

只有父类的私有属性和方法是不能被子类继承的！！！

- 子类皆父类

继承注意事项：

1.java中只支持单继承，也就是只能继承一个直接的父类

2.java中所有的类都是object的子类

3.在开发中使用继承是很有必要的，这样可以更好的使用别人写好的类

##### 方法的重载：

简单的说：方法重载就是类的同一种功能的多种实现方式，到底采用哪种方式，取决于调用者给出的参数

方法重载注意：

它的范围是在同一个类中定义不同的实现形式

1.方法名相同（便于记忆，比如我要定义一个比较两个数，返回最大数的功能，只用一个getMax()即可，其他整型，浮点型的实现也是用这个方法名）

2.如果仅仅是方法的返回值类型不同，是不能构成重载的！！！（即使后面参数都相同）

3.方法的参数类型、个数、顺序必须至少有一项不同

4.方法的修饰符和返回值可以不同

5.如果仅仅是控制访问修饰符号不一样也是不能构成重载的！！！

##### 方法的覆盖：范围，子类和父类之间

子类重写父类的方法！！！

子类中的方法和父类中的完全一样

如果要想权限不一样，也只能是权限越来越宽松

#### 抽象类：

abstract class Animal{

abstract public void cry();//体现类的设计者设计思想

}

抽象类的必要性：

抽象类中的抽象方法没有方法体{}

继承抽象父类，必须实现父类的所有抽象方法

抽象类中可以有普通方法

接口就是类的功能的抽象，是方法的抽象

interface Usb{

public void start();

public void stop();

}

接口中的所有方法默认就是抽象方法，实现接口必须实现他的所有方法

有很多的类来实现这个接口，但是不同的类的实现形式是不一样的，也就是接口方法的方法体不一样！！！

接口就是有多态的感觉，接口其实也体现了多态，多态就是：函数的参数是一个父类，这样所有的子类都可以作为参数传到这个函数，因为子类皆父类

注意：接口不可以实例化

接口中的所有方法都不能有方法体抽象类可以有

所以接口是更加抽象的抽象类

一个类可以实现多个接口 逗号隔开

这个类要实现所有接口的所有方法

接口中可以有变量，但是必须是public的，默认就是静态的、final的，不能是其他的

interface Usb{

int a=1;//必须初始化

}

所以接口里面的变量，默认就是 public、static、final

即是可以直接通过接口名来访问接口的变量

接口可以继承别的接口（不能继承类）

interface a{}

interface b extends a{}

class pc implements b{

//这里要实现a、b接口的所有方法！！！

}

接口体现了程序设计的多态，高内聚低耦合（也就是很分散）

继承父类和实现接口是可以同时的

eg：class Phone extends Telphone implements Usb{

}

Phong 继承了 Telphone 实现的 Usb 接口

理解继承和接口：继承是直接得到父类的所有可见属性，功能；实现接口就相当于后天去学习一项新的技能，必须靠自己来重新学习

实现接口是对java单继承的一种补充

继承是有层级关系的，牵一发而动全身，不是很灵活，而接口是很灵活的，可以在不打破类的平衡关系的基础上给类添加新的功能

前期绑定和后期绑定：

前期绑定：在编译时就能确定变量的类型

后期绑定：编译时不知道变量是什么，只有在运行过程中才确定，比如多态

final：最终版本，不允许子类修改，子类可继承 可应用于单例模式

java一般这样命名final变量

final aaa\_bbb\_ccc = 44.4f;

int a;默认值为0，最好赋初值

final 定义的变量必须赋初值

final修饰的类不允许其他类继承！！！

eclipse中的编译器只支持1.7版本

如何解决？可以到eclipsemarketplace下载java8

安装这个插件：

-----------------------------------------

作业测试题总结：

case 1： //这里必须是常量，不可以是变量

byte a = 1;

switch(a){

case 1/1://可以，低精度可以自动转成高精度 1/1 为int型

}

switch 和 case 搭配，哪个case语句符合，就执行哪一段，如果没有bread就继续往下执行

可以这样定义一个变量 int a,b;

在for循环中，可以for(int a=1,b=2;;)

构造函数跟其他成员方法一样都是可以用各种权限修饰符进行修饰，只不过构造函数在创建对象时就自动调用

在java中 逗号 , 表示并列

数组：一维 二维 排序 查找

快速排序难度高，效率高

数组的声明：

1. int arr[];

2. int[] arr;

以上两种方法都可以

数组的创建：

arr = new int[5];

数组的赋值和访问：

arr[0] = 0;

arr[0];

在java中数组是一个对象，用new来创建

可以用for循环来遍历数组

for(int i=0;i<6;i++){

System.out.println(arr[i]);

}

数组的长度 arr.length，当不知道数组长度时，可以用这个！！！

声明和赋值同时进行

int arr[] ={1,2,3,45,6};

这里不用指定数组长度，他会自动根据内容计算，多少都可以存的进去

int float double 这些简单的数据类型可以有数组，对象也可以有数组

比如有一个Dog类

Dog dogs[] = new Dog[4];

注意：这里的dogs[2];存放的是一个地址，用来指向一个真正的对象，刚开始的时候为 null ，所以我们还要让这个数组单元指向一个对象才可以

不可以直接 dogs[1].getName(); xxx

应该这样：

dogs[0] = new Dog();

dogs[0].getName();

-----------------------------------------

对数组的各种操作

1.找最大值

打擂法，先假设第一个为最大的，在for循环中，从第二个开始，与最大的数比较，把较大的放到假设的那个位置（也就是擂主），以此类推，就是每一个数都与当前的擂主比拼 ...最后留下来的就是最终擂主

需要注意的是，下标也要相应改变为擂主的下标（一开始为0）

在比较数 int double 时可以使用 ==

在java中 == 比较的是两个字符串的地址，如果要比较两个字符串的内容必须要用equal()方法 ，比如：

java中数组的大小必须事先指定

也就是 ：int arr[] = new int[5];

int arr[] = {1,2,4,3,5};

这两种方法都指定了数组大小

排序的分类：

1.内部排序

将所有要排序的数据都加入到内存中进行排序，分为：交换式排序、选择式排序、

2.外部排序

数据量很大时，无法全部加载到内存中，需要借助外部存储排序

交换式排序：冒泡排序、快速排序

冒泡（比较慢）：

技巧：

类的前面定义的变量称为字段，可以在eclipse中利用字段生成get和set方法，也可以通过get、set方法自动生成字段，这就是自动修正