## 类和对象

章铁飞 浙江工商大学

tfzhang@mail.zjgsu.edu.cn

### 说明：

**1、要自学的两部分内容：**

请自学how2j中的[JAVA 基础](http://how2j.cn/stage/12.html) / [面向对象](http://how2j.cn/module/76.html) /：

http://how2j.cn/k/oo/oo-class-object/286.html

和[JAVA 基础](http://how2j.cn/stage/12.html) / [类和对象](http://how2j.cn/module/53.html) / ：http://how2j.cn/k/class-object/class-object-reference/307.html后，完成下面的练习题。

**2、答题格式：**

每个练习题下面请粘贴对应的源代码，如果程序要求有输出，则对输出进行截图并且附在源代码下面。

举例：

给Student类增加一个void speak()方法，无参数，无返回。调用结果是输出如下的语句：

I can speak Chinese and English. 书写代码分别调用zhang3和li4的speak方法。

源代码：

**public** **class** Student {

String name;

**int** height;

**int** weight;

**public** **void** speak(){

System.***out***.println("I can speak language");

}

**public** **static** **void** main(String[] args){

Student zhang3 = **new** Student();

Student li4 = **new** Student();

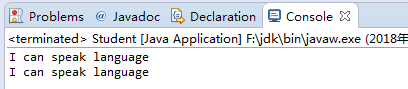
zhang3.speak();

li4.speak();

}

}

输出：



**3. 作业提交时间和地址：**

提交本word文档，并且命名为：班级+学号+姓名+第5周作业.docx

提交的地址：ftp://10.21.11.21/学生作业上传/章铁飞/Java程序设计2018/第5周-第2次作业/

截止时间：下次课之前

### 题目：

0. 查阅官方文档：已经知道Arrays.sort(a)是将数组a按升序排序，请查阅官方文档，找出Arrays将数组进行降序排序的方法，并写出测试代码。

源代码：

import java.util.\*;

public class Sort {

public static void main(String args[]) {

Integer[] a = {121, 31, 65, 165, 561, 56, 165, 156, 88, 275, 485};

Comparator cmp = new MyComparator();

Arrays.sort(a, cmp);

for (int i : a) {

System.out.println(i+' ');

}

}

}

class MyComparator implements Comparator<Integer>{

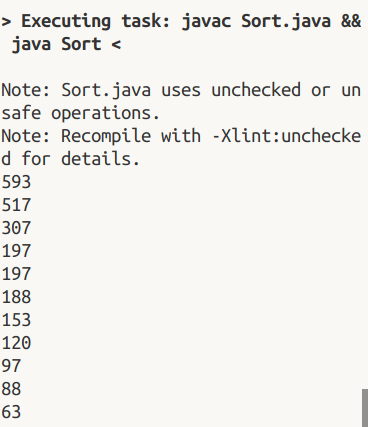
public int compare(Integer o1, Integer o2) {

return o2-o1;

}

}

输出：



1. 参照Hero类，设计一个Student类，属性包含姓名、身高和体重，写出对应的Student类代码。

源代码：

public class Student {

String name;

double height;

double weight;

public static void main(String[] args) {

}

}

2. 创建两个具体的学生，名字分别为zhang3和li4，身高分别为1.65和1.80，体重为56(kg)和60(kg)，请写出对应的代码。

源代码：

public class Student {

String name;

double height;

double weight;

public static void main(String[] args) {

Student zhang3 = new Student();

zhang3.name = "zhang3";

zhang3.height = 1.65;

zhang3.weight = 56;

Student li4 = new Student();

li4.name = "li4";

li4.height = 1.80;

li4.weight = 60;

}

}

3. 给Student类增加一个void speak()方法，无参数，无返回。调用结果是输出如下的语句：

I can speak language. 书写代码分别调用zhang3和li4的speak方法。

源代码：

public class Student {

String name;

double height;

double weight;

public void speak(){

System.out.println("I can speak language");

}

public static void main(String[] args) {

Student zhang3 = new Student();

zhang3.speak();

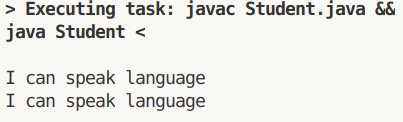
Student li4 = new Student();

li4.speak();

}

}

输出：



4. 给Student增加一个private的属性score，表示期末分数。并且增加两个方法：

a. void setScore(int s)：设定score的值；

b. int getScore(): 返回设定score的值；

将zhang3的score设置为78，并且再通过getScore取出打印出来。请写出相关可以正常运行的代码，并将源代码和输出的截图粘贴此处。

源代码：

public class Student {

String name;

double height;

double weight;

private int score;

public void setScore(int s) {

this.score = s;

}

public int getScore() {

return this.score;

}

public static void main(String[] args) {

Student zhang3 = new Student();

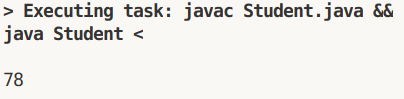
zhang3.setScore(78);

System.out.println(zhang3.getScore());

}

}

输出：



5. 类的继承：请创建一个People类，包含属性姓名，身高，体重。已经Student类的属性包含：姓名、身高、体重、分数。请使用继承的方式，实现Student类，并且在main方法中创建一个具体的学生zhang3。

源代码：

public class Student extends People {

private int score;

public static void main(String[] args) {

Student zhang3 = new Student();

zhang3.name = "zhang3";

zhang3.height = 1.65;

zhang3.weight = 56;

zhang3.score = 78;

}

}

public class People {

String name;

double height;

double weight;

public static void main(String[] args) {

}

}

6. 类方法的重载：

给Student类增加如下的三种speak方法：

void speak(); 输出"I can speak nothing"

void speak(String lang1); 如果lang1=Chinese，则输出"I can speak Chinese"

void speak(String lang1, String lang2); 如果lang1=Chinese, lang2=English, 则输出"I can speak Chinese and English"

创建Student实例zhang3，并且依次调用上述三种speak方法。

源代码：

public class Student {

public void speak() {

System.out.println("I can speak nothing");

}

public void speak(String lang1) {

System.out.println("I can speak "+lang1);

}

public void speak(String lang1, String lang2) {

System.out.println("I can speak "+lang1+" and "+lang2);

}

public static void main(String[] args) {

Student zhang3 = new Student();

zhang3.speak();

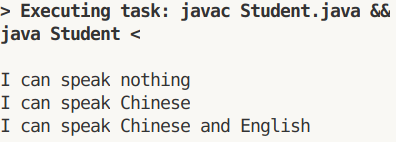
zhang3.speak("Chinese");

zhang3.speak("Chinese", "English");

}

}

输出：



7. 类的构造方法：

给Student类创建一个构造方法public Student()（方法中调用println语句输出点东西），main方法中new出一个实例，观察构造方法的被调用情况。再给Student类再创建一个有参的构造函数public Student(String name)，完成姓名的初始化，同样地，main方法中new出一个实例，观察有参构造方法的调用情况。

源代码：

public class Student {

public Student() {

System.out.println("I have no name");

}

public Student(String name) {

System.out.println("My name is "+name);

}

public static void main(String[] args) {

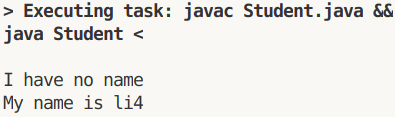
Student zhang3 = new Student();

Student li4 = new Student("li4");

}

}

输出：



8. this关键词：

Student类中，增加如下的方法：

public void showThis(){

System.out.println(this);

}

main方法中new一个zhang3，然后调用zhang3.showThis()观察输出。

在main方法中，然后直接使用如下的语句：System.out.println(zhang3);

对比两者的输出。

源代码：

public class Student {

public void showThis(){

System.out.println(this);

}

public static void main(String[] args) {

Student zhang3 = new Student();

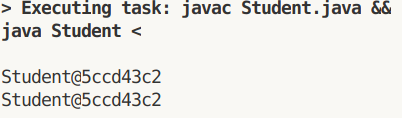
zhang3.showThis();

System.out.println(zhang3);

}

}

输出：



9. Student类中添加如下的assist方法：

public void assist(Student st, int addScore){

st.score += addScore;

}

表示对学生对象st辅导以后，帮助他提高了分数。假设有zhang3和li4两个学生，zhang3辅导li4提高10分:zhang3.assist(li4, 10);

源代码：

public class Student {

private int score;

public void setScore(int s) {

this.score = s;

}

public int getScore() {

return this.score;

}

public void assist(Student st, int addScore){

st.score += addScore;

}

public static void main(String[] args) {

Student zhang3 = new Student();

Student li4 = new Student();

zhang3.setScore(95);

li4.setScore(80);

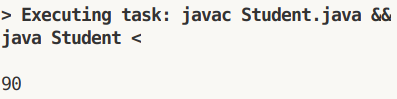
zhang3.assist(li4, 10);

System.out.println(li4.getScore());

}

}

输出：



10. 单例模式：

创建一个班级Banji类，其中包含属性变量className，并且使用饿汉式单例模式. 写出对应的代码，并且测试单例模式是否正常工作。

public class Banji {

String className;

private Banji(){

}

private static Banji jike1701 = new Banji();

public static Banji getclass(){

return jike1701;

}

public static void main(String[] args) {

Banji c1 = Banji.getclass();

Banji c2 = Banji.getclass();

Banji c3 = Banji.getclass();

System.out.println(c1==c2);

System.out.println(c1==c3);

}

}

输出：

