**类和对象作业**

1. **填空题**
2. 类是组成Java程序的基本要素，类体有两部分构成：一部分是变量的定义，另一部分是 **方法**  的定义。
3. 执行Person p = new Person();语句后，将在\_\_\_**堆内存**\_\_中给Person对象分配空间，并在栈内存中给引用变量p分配空间，存放Person对象的引用。
4. **构造方法** 是一种特殊方法，它的名字必须与它所在的类的名字完全相同，并且不书写返回值类型，在创建对象实例时由new运算符自动调用。
5. 局部变量的名字与成员变量的名字相同，若想在该方法内使用成员变量，必须使用关键字 **this** 。
6. 使用关键字 **this** 来调用同类的其它构造方法，优点同样是以最大限度地代码的利用程度 ，减少程序的维护工作量。
7. 用关键字 **static** 修饰的成员变量是类变量，类变量是指不管类创建了多少对象，系统仅在第一次调用类的时候为类变量分配内存，所有对象共享该类的类变量。
8. 使用static修饰的变量称为静态变量，静态变量可以有两种访问方式，分别是类名.静态变量名和\_\_\_\_**对象名.静态变量名**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_。
9. 在一个类文件中的关键字package，import，class出现的可能顺序是\_\_**package，import，class**\_\_\_\_\_。
10. **package**  关键字作为Java源文件的第一条非注释性语句，指明该源文件定义的类所有的包。
11. **java.lang** 包是Java语言的核心类库，它包含了运行Java程序必不可少的系统类，使用该包下的类和接口不需要使用import导入。
12. **选择题**

|  |  |  |
| --- | --- | --- |
| **1.** | **在Java中，以下程序编译运行后的输出结果为（ D）。（选择一项）** | |
|  | **public** **class** Test {  **int** x, y;  Test(**int** x, **int** y) {  **this**.x = x;  **this**.y = y;  }  **public** **static** **void** main(String[] args) {  Test pt1, pt2;  pt1 = **new** Test(3, 3);  pt2 = **new** Test(4, 4);  System.***out***.print(pt1.x + pt2.x);  }  } | |
|  |  |  |
|  | **A** | 6 |
|  | **B.** | 34 |
|  | **C.** | 8 |
|  | **D.** | 7 |

|  |  |  |
| --- | --- | --- |
| **2.** | **分析如下Java程序的代码所示，则编译运行后的输出结果是（ C ）。（选择一项）** | |
|  | **public** **class** Test {  **int** **count**=9;  **public** **void** count1(){  **count** =10;  System.*out*.println("count1="+ **count**);  }  **public** **void** count2(){  System.*out*.println("count2="+ **count**);  }  **public** **static** **void** main(String[ ] args) {  Test t=**new** Test();  t.count1();  t.count2();  }  } | |
|  |  |  |
|  | **A** | count1=9;  count2=9; |
|  | **B.** | count1=10;  count2=9; |
|  | **C.** | count1=10;  count2=10; |
|  | **D.** | count1=9;  count2=10; |

|  |  |  |
| --- | --- | --- |
| **3.** | **以下语句中关于Java构造方法的说法错误的是（ B ）。（选择一项）** | |
|  |  |  |
|  | **A.** | 构造方法的作用是为创建对象进行初始化工作，比如给成员变量赋值 |
|  | **B.** | 一个Java类可以没有构造方法，也可以提供1个或多个构造方法 |
|  | **C.** | 构造方法与类同名，不能书写返回值类型 |
|  | **D.** | 构造方法的第一条语句如果是super()，则可以省略，该语句作用是调用父类无参数的构造方法 |

|  |  |  |
| --- | --- | --- |
| **4.** | **在Java中关于静态方法，以下说法中正确的是（ AC ）。（选择两项）** | |
|  |  |  |
|  | **A** | 静态方法中不能直接调用非静态方法 |
|  | **B.** | 非静态方法中不能直接调用静态方法 |
|  | **C.** | 静态方法可以用类名直接调用 |
|  | **D.** | 静态方法里可以使用this |

|  |  |  |
| --- | --- | --- |
| **5.** | **下列选项中关于Java中类方法的说法错误的是（ A C ）。（选择二项）** | |
|  |  |  |
|  | **A** | 在类方法中可用this来调用本类的类方法 |
|  | **B.** | 在类方法中调用本类的类方法时可直接调用 |
|  | **C.** | 在类方法中只能调用本类中的类方法 |
|  | **D.** | 在类方法中调用实例方法需要先创建对象 |

1. **判断题**
2. 类可以看成一类对象的模板，对象可以看成该类的一个具体实例。( **〇** )
3. 如果没有为类中的某些成员赋初始值，Java会为类成员赋予固定的初始值，如数值变量的值为0，布尔变量的值为true，未初始化的引用为null。（ **×** ）
4. Java中所有的变量，不管是成员变量还是局部变量，在使用前都必须进行初始化。（ **×** ）
5. 在Java中对象可以赋值，只要使用赋值运算符即可，相当于生成了一个各属性与赋值对象相同的新对象。（ **×** ）
6. System.out.println(“Hello java!”)中out是System类的一个静态成员变量。（ **〇** ）
7. 构造方法用于创建对象，一般不允许有任何返回值，因此需要在构造方法返回类型处标注为void。（ **×** ）
8. Java语言为所有的Java程序自动导入包“java.lang”，因此Java程序可以直接用“java.lang”中的类和接口。（ **〇** ）
9. **简答题**
   1. 面向过程和面向对象的区别。（5分）
   2. 类和对象的关系
   3. 构造方法的作用和特征
   4. this关键字的作用和用法
   5. 简述static关键字的作用。

提示：从static可以修饰变量，方法，代码块，内部类四个方面来回答。（5分）

1. **编码题**
2. 请定义一个交通工具(Vehicle)的类其中有: 属性： 速度(speed)、 体积(size)等，方法：移动(move())、设置速度(setSpeed(int speed))、加速 speedUp()、减速 speedDown()等。最后在测试类 Vehicle 中的 main()中实例化一个交通工具对象并通过方法给它初始化 speed,size 的值并且通过打印出来。另外调用加速、减速的方法对速度进行改变。

**public class** Vehicle {  
 **public static void** main(String[] args) {  
 Vehicle transport = **new** Vehicle();  
 transport.move();  
 transport.setSpeed(50);  
 transport.setSize(100);  
 transport.speedUp(10);  
 transport.speedDown(20);  
 }  
  
 **int speed**=0;  
 **int size**=0;  
  
 **public void** move(){  
 System.***out***.println(**"车在移动"**);  
 }  
  
 **public void** setSpeed(**int** speed){  
 **this**.**speed** = speed;  
 System.***out***.println(**"速度为："**+speed+**"km/h"**);  
 }  
  
 **public void** setSize(**int** size){  
 **this**.**size** = size;  
 System.***out***.println(**"体积为："**+size+**"吨"**);  
 }  
  
 **public void** speedUp(**int** add){  
 **this**.**speed** += add;  
 System.***out***.println(**"速度增加"**+add+**"之后的速度为："**+**speed**+**"km/h"**);  
 }  
  
 **public void** speedDown(**int** minuse){  
 **this**.**speed** -= minuse;  
 System.***out***.println(**"速度减少"**+minuse+**"之后的速度为："**+**speed**+**"km/h"**);  
 }  
}

1. 编写 Java 程序用于显示人的姓名和年龄。定义一个人类Person。 该类中应该有两个私有属性： 姓名 （name） 和年龄 （age） 。定义构造方法用来初始化数据成员。再定义显示（display()） 方法将姓名和年龄打印出来。在 main 方法中创建人类的实例然后将信息显示。

**public class** Person {  
  
 String **name**;  
 **int age**;  
  
 **public** Person(String name,**int** age){  
 **this**.**name** = name;  
 **this**.**age** = age;  
 }  
  
 **public void** display(){  
 System.***out***.println(**"姓名为："**+**this**.**name**+**",年龄为："**+**this**.**age**);  
 }  
  
 **public static void** main(String[] args) {  
 Person person1 = **new** Person(**"ling"**,23);  
 person1.display();  
 }  
}

1. 定义一个圆类——Circle，在类的内部提供一个属性：半径(r)，同时 提供 两个 方 法 ： 计算 面积 （ getArea() ） 和 计算 周长（getPerimeter()） 。 通过两个方法计算圆的周长和面积并且对计算结果进行输出。最后定义一个测试类对 Circle 类进行使用。
2. **public class** Circle {  
     
    **int r**;  
    **double pai** = 3.14;  
     
    **public** Circle(**int** r){  
    **this**.**r** = r;  
    }  
     
    **public void** getArea(){  
    **double** area = **pai**\*Math.*pow*(**r**,2);  
    System.***out***.println(**"面积为:"**+area);  
    }  
     
    **public void** getPerimeter(){  
    **double** perimeter = 2\***pai**\***r**;  
    System.***out***.println(**"周长为:"**+perimeter);  
    }  
     
    **public static void** main(String[] args) {  
    Circle circle = **new** Circle(5);  
    circle.getArea();  
    circle.getPerimeter();  
    }  
   }
3. 构造方法与重载：定义一个网络用户类，要处理的信息有用户 ID、用户密码、 email 地址。在建立类的实例时把以上三个信息都作为构造函数的参数输入， 其中用户 ID 和用户密码时必须缺省时 email地址是用户 ID 加上字符串"@gameschool.com"。
4. **public class** People {  
     
    String **ID**;  
    String **password**;  
    String **emailAddress**;  
     
    **public** People(){  
     
    }  
     
    **public** People(String ID,String password,String emailAddress){  
    **this**.**ID** = ID;  
    **this**.**password** = password;  
    **this**.**emailAddress** = emailAddress;  
    }  
     
    **public** People(String ID,String password){  
    **this**.**ID** = ID;  
    **this**.**password** = password;  
    **this**.**emailAddress** = **this**.**ID**+**"@gameschool.com"**;  
    }  
     
    **public void** output(){  
    System.***out***.println(**"ID为:"**+**ID**);  
    System.***out***.println(**"密码为:"**+**password**);  
    System.***out***.println(**"email为:"**+**emailAddress**);  
    }  
     
    **public static void** main(String[] args) {  
    People people1 = **new** People(**"123"**,**"abc"**,**"ling@gameschool.com"**);  
    People people2 = **new** People(**"456"**,**"def"**);  
    people1.output();  
    people2.output();  
    }  
   }
5. **可选题**
   1. 定义一个类 Calculaion， 其中包含四个方法： 加（add()） 、 减（sub()） 、乘（times()） 和除（div()） 。 创建一个具有 main()函数的类。 在 main()函数中创建一个 Calculation 的实例对象并对其中的方法进行调用。

**public class** Calculation {  
  
 **double a**;  
 **double b**;  
  
 **public** Calculation(){  
 }  
  
 **public** Calculation(**double** a,**double** b){  
 **this**.**a** = a;  
 **this**.**b** = b;  
 }  
  
 **public void** add(){  
 System.***out***.println(**"a和b相加的结果为："**+(**a**+**b**));  
 }  
  
 **public void** sub(){  
 System.***out***.println(**"a和b相减的结果为："**+(**a**-**b**));  
 }  
  
 **public void** times(){  
 System.***out***.println(**"a乘b ："**+(**a**\***b**));  
 }  
 **public void** div(){  
 System.***out***.println(**"a除以b ："**+(**a**/**b**));  
 }  
  
 **public static void** main(String[] args) {  
 Calculation number = **new** Calculation(8,2);  
 number.add();  
 number.sub();  
 number.times();  
 number.div();  
 }  
}

* 1. 定义一个类 Draw ， 在 类 中 提供 3 个 方法：输出直角三角形（drawTrian()） 、 输出矩形（drawRec()） 及平行四边形（drawPra()） 。通过方法可以输出由“\*” 组成的一个图形。 同时在类中包含两个属性：星号的个数（count） 、行数（lines） 。最后在测试类中进行调用。

**public class** Draw {  
 **int count**;  
 **int lines**;  
  
 **public** Draw(){  
 }  
  
 **public** Draw(**int** count,**int** lines){  
 **this**.**count** = count;  
 **this**.**lines** = lines;  
 }  
  
 **public void** drawTrian(){  
 **if**(**count**>=**lines**){  
 **for**(**int** i=1;i<=**lines**;i++){  
 **for**(**int** j=1;j<=i;j++){  
 System.***out***.print(**"\*"**);  
 }  
 System.***out***.println();  
 }  
 }**else if**(**count**<**lines**){  
 **for**(**int** i=1;i<=**count**;i++){  
 **for**(**int** j=1;j<=i;j++){  
 System.***out***.print(**"\*"**);  
 }  
 System.***out***.println();  
 }  
 }  
  
 }  
  
 **public void** drawRec(){  
 **for**(**int** i=1;i<=**lines**;i++){  
 **for**(**int** j=1;j<=**count**;j++){  
 System.***out***.print(**"\*"**);  
 }  
 System.***out***.println();  
 }  
 }  
  
 **public void** drawPra(){  
 **for**(**int** i=1;i<=**lines**;i++){  
 **for**(**int** j=1;j<=i-1;j++){  
 System.***out***.print(**" "**);  
 }  
 **for**(**int** k=1;k<=**count**;k++){  
 System.***out***.print(**"\*"**);  
 }  
 System.***out***.println();  
 }  
 }  
  
 **public static void** main(String[] args) {  
 Draw trian = **new** Draw(9,4);  
 **//trian.drawTrian();  
 //trian.drawRec();** trian.drawPra();  
  
 }  
  
}

* 1. 创建一个空调，并调用制冷、制热、通风功能。空调包含的属性有品牌、匹数、温度，功能有加热、制冷、通风等功能。

**public** **class** AirConditioning {

// 空调属性

String name;

String pinpai;

String hp;// 匹数

**int** tempeture;

// 定义制冷行为

**void** cool() {

tempeture--;

}

// 定义加热行为

**void** hot() {

tempeture += 7;

}

//通风功能

**void** ventilate(){

System.***out***.println("空气通畅，温度不变");

}

// 主函数

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

AirConditioning kt = **new** AirConditioning();

kt.hp = "2匹";

kt.tempeture = 20;

kt.pinpai = "格力";

kt.cool();

System.***out***.println("空调的温度是" + kt.tempeture);

kt.hot();

System.***out***.println("空调的温度是" + kt.tempeture);

kt.ventilate();

}

}

* 1. 定义一个表示学生信息的类 Student，要求如下：

（1）类 Student 的属性如下：

sNO 表示学号； sName 表示姓名； sSex 表示性别； sAge 表示

年龄；sJava：表示 Java 课程成绩。

（2）类 Student 带参数的构造方法：

在构造方法中通过形参完成对成员变量的赋值操作。

（3）类 Student 的方法成员：

getNo（）：获得学号；

getName（）：获得姓名；

getSex（）：获得性别；

getAge（）获得年龄；

getJava（）：获得 Java 课程成绩

（4） 根据类 Student 的定义，创建五个该类的对象，输出每个学生的信息，计算并输出这五个学生 Java 语言成绩的平均值，以及计算并输出他们 Java 语言成绩的最大值和最小值。

**public class** Student {  
  
 String **sNO**;  
 String **sName**;  
 String **sSex**;  
 **int sAge**;  
 **int sJava**;  
 **int arr**[] ;  
  
 **public** Student(){  
 }  
  
 **public** Student(String sNO, String sName, String sSex, **int** sAge, **int** sJava) {  
 **this**.**sNO** = sNO;  
 **this**.**sName** = sName;  
 **this**.**sSex** = sSex;  
 **this**.**sAge** = sAge;  
 **this**.**sJava** = sJava;  
 }  
  
 **public** String getNo(){  
 **return sNO**;  
 }  
  
 **public** String getName(){  
 **return sName**;  
 }  
  
 **public** String getSex(){  
 **return sSex**;  
 }  
  
 **public int** getAge(){  
 **return sAge**;  
 }  
  
 **public int** getJava(){  
 **return sJava**;  
 }  
  
**// public static int getAverage(Student arr[]){  
// double sum = 0;  
// for(int i=0;i<arr.length;i++){  
// sum+=arr[i].sJava;  
// }  
// return (int)(sum/arr.length);  
// }  
  
  
 public static int** getAverage2(**int** arr[]){  
 **double** sum = 0;  
 **for**(**int** i=0;i<arr.**length**;i++){  
 sum+=arr[i];  
 }  
 **return** (**int**)(sum/arr.**length**);  
 }  
  
 **public static double** getMax(**int** arr[]){  
 **double** max = -1;  
 **for** (**int** i = 0; i<arr.**length** ; i++) {  
 **if**(arr[i]>max){  
 max = arr[i];  
 }  
 }**return** max;  
 }  
  
 **public static double** getMin(**int** arr[]){  
 **double** min = 101;  
 **for** (**int** i = 0; i<arr.**length** ; i++) {  
 **if**(arr[i]<min){  
 min = arr[i];  
 }  
 }**return** min;  
 }  
  
 **public static void** main(String[] args) {  
 Student s1 = **new** Student(**"1"**,**"ling"**,**"女"**,28,100);  
 Student s2 = **new** Student(**"2"**,**"gen"**,**"男"**,25,80);  
 Student s3 = **new** Student(**"3"**,**"lala"**,**"女"**,26,90);  
 Student s4 = **new** Student(**"4"**,**"juan"**,**"女"**,21,100);  
 Student s5 = **new** Student(**"5"**,**"feng"**,**"男"**,28,100);  
  
**// Student arr[] = new Student[]{s1,s2,s3,s4,s5};  
// int average = getAverage(arr);  
// System.out.println(average);  
  
 int** arr[] = **new int**[]{s1.**sJava**,s2.**sJava**,s3.**sJava**,s4.**sJava**,s5.**sJava**};  
 **int** average = *getAverage2*(arr);  
 System.***out***.println(**"平均分为："**+average);  
 **double** max = *getMax*(arr);  
 System.***out***.println(**"最高分为："**+max);  
 **double** min = *getMin*(arr);  
 System.***out***.println(**"最低分为："**+min);  
 }  
}

※上面这道题里注释的部分是成对的，也可以实现求平均分