一、单项选择题（每题 2 分，共 20 分）

1、Java 语言是(　D　)

A、面向问题的解释型高级编程语言

B、面向机器的低级编程语言

C、面向过程的编译型高级编程语言

D、面向对象的解释型高级编程语言

2、下列哪个是合法的 Java 标识符(　A　)?

A．&2　　B．123.9　　C．_2#　　D．public

3、编译 Java Application 源程序文件产生的字节码文件的扩展名为(B　)。

A．java　　B．class　　C．html　　D．exe

4、有关类 Demo，哪句描述是正确的（　A　）？

```
public class Demo extends Base{
    private int count;
    public Demo(){
            System.out.println("A Demo object has been created");
    }
    protected void addOne() {count++; }
}
```

A．当创建一个 Demo 类的实例对象时，count 的值为 0。

B．当创建一个 Demo 类的实例对象时，count 的值是不确定的。

C．超类对象中可以包含改变 count 值的方法。

D．Demo 的子类对象可以访问 count。

5、Java Application 源程序的主类是指包含有(　A　)方法的类。

A、main 方法　　B、toString 方法　　C、init 方法　　D、actionPerformed 方法

6、如果任何包中的子类都能访问超类中的成员，那么应使用哪个限定词？（C）

A．public　　B．private　　C．protected　　D．transient

7、在 Java 中，存放字符串常量的对象属于(　B　)类对象。

A、Character　　B、String　　C、StringBuffer　　D、Vector

8、在使用 interface 声明一个接口时，只可以使用(　D　)修饰符修饰该接口。

A、private　　B、protected　　C、private protected　　D、public

9、在编写异常处理的 Java 程序中，每个 catch 语句块都应该与(　C　)语句块对应，使得用该语句块来启动 Java 的异常处理机制。

A．if - else　　B．switch　　C．try　　D．throw

10、以下由 do-while 语句构成的循环执行的次数是(　B　)

```
int k = 0;
do { ++k; }while ( k < 1 );
```

A、一次也不执行　　B、执行 1 次

C、无限次　　D、有语法错，不能执行

二、填空（每空 2 分，共 30 分）

1、在 Java 中有两种多态，一种是使用方法的__重载_____实现多态，另一种是使用方法的____重写____实现多态。

2、在 Java 程序中，通过类的定义只能实现_____单_____重继承，但通过接口的定义可以实现____多___重继承关系。

3、设 x = 2 ，则表达式 (x + +) * 3 的值是_____。

答：6 **简单 java 程序的考查**

4、若 x = 5，y = 10，则 x > y 和 x <= y 的逻辑值分别为_____和_____。

答：false、true

5、Java 中所有类都是类_____的子类。

答：Ojbject

6、一个 Java Application 源程序文件名为 MyJavaApplication.java，如果使用 Sun 公司的 Java 开发工具 JDK 编译该源程序文件并使用其虚拟机运算这个程序的字节码文件，应该顺序执行如下两个命令：_____、_____。

答：javac MyJavaApplication.java 、java MyJavaApplication

7、_____方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。_____方法是不能被当前类的子类重新定义的方法。**对于方法的理解。**

答：抽象方法、最终方法 (或 abstract 方法、final 方法)

8、如果类中的成员变量只能被该类中的方法访问或引用，则该变量应该用_____ 修饰 **类与修饰的考查**

答：private

9、 字符串分为两大类，一类是字符串常量，使用_____类的对象表示；另一类是的字符串变量，使用_____类的对象表示。**关于如何定义对象的考查**

答：String、StringBuffer

三、判断题（每题 2 分，共 20 分）

1、for 语句中的循环体不能为空。（ 错 ）

2、接口是由常量和抽象方法组成的特殊类。（ 对 ）

3、构造函数的方法名可由编程人员任意命名。（ 错 ）

4、类的私有属性和私有方法可以被其子类访问。（ 错 ）

5、因为 Java 不支持多重继承，所以定义类时 implements 关键字后面只能说明一个接口名。（ 错 ）

6、abstract 是抽象修饰符，可以用来修饰类及其属性和方法。（ 对 ）

7、一个 Java 源程序中允许有多个公共类。（ 错 ）

8、一个 catch 块也可以区分处理多个不同类型的异常，只要它们是该 catch 语句块异常参数的子类或其本身。（ 对 ）

9、程序中一旦执行了 catch 语句块，则不会执行 finally 语句块。（ 错 ）

10、引用一个类的属性或调用其方法，必须以这个类的对象为前缀。（ 错 ）

四、阅读程序，写出运行结果（每题 5 分，共 10 分）

五、程序设计（20 分）

（1）编写一个圆类 Circle，该类拥有：

①一个成员变量

Radius（私有，浮点型）； // 存放圆的半径；

②两个构造方法

Circle( ) // 将半径设为 0

Circle(double r) //创建 Circle 对象时将半径初始化为 r

③ 三个成员方法

double getArea( ) //获取圆的面积

double getPerimeter( ) //获取圆的周长

void show( ) //将圆的半径、周长、面积输出到屏幕

（2）编写一个圆柱体类 Cylinder，它继承于上面的 Circle 类。还拥有：

①一个成员变量

double hight（私有，浮点型）； // 圆柱体的高；

②构造方法

Cylinder (double r, double h) //创建 Circle 对象时将半径初始化为 r

③ 成员方法

double getVolume( ) //获取圆柱体的体积

void showVolume( ) //将圆柱体的体积输出到屏幕

（3）编写应用程序，创建类的对象，分别设置圆的半径、圆柱体的高，计算并分别显示圆半径、圆面积、圆周长，圆柱体的体积。

```
class Circle {                          //定义父类--园类
      private double radius;            //成员变量--园半径
      Circle() {                        //构造方法
            radius=0.0;
      }
      Circle(double r) {                //构造方法
            radius=r;
      }
      double getArea() {                //成员方法--求园面积
            return Math.PI*radius*radius;
      }
      double getPerimeter() {           //成员方法--求园周长
```

```
            return 2*Math.PI*radius;
        }
        void show() {                       //成员方法--显示园半径、周长、面积
            System.out.println("园半径="+radius);
            System.out.println("园周长="+getPerimeter());
            System.out.println("园面积="+getArea());
        }
}
class Cylinder extends Circle {      //定义子类--圆柱类
        private double hight;               //成员变量--园柱高
        Cylinder(double r,double h) {    //构造方法
            super(r);
            hight=h;
        }
        public double getVol() {          //成员方法--求园柱体积
            return getArea()*hight;
        }
        public void showVol() {           //成员方法--显示园柱体积
            System.out.println("圆柱体积="+getVol());
        }
}
public class vvv {          //定义主类
public static void main(String[] args) {     //主程入口
    Circle Ci=new Circle(10.0); // 生成园类实例
    Ci.show(); // 调用园类的方法
        Cylinder Cyl=new Cylinder(5.0,10.0);     //生成圆柱类实例
        Cyl.show();                             //调用父类方法
        Cyl.showVol();                          //调用子类方法
    }
}
```