## Java综合测试题

单选或多选

1、JAVA所定义的版本中不包括：（ BD ）

JAVA2 EE

JAVA2 Card

JAVA2 ME

JAVA2 HE

JAVA2 SE

2、下列说法正确的是（ AC ）

JAVA程序的main方法必须写在类里面

JAVA程序中可以有多个main方法

JAVA程序中类名必须与文件名一样 //当此类不是用public来修饰的时，可以不一样。

JAVA程序的main方法中如果只有一条语句，可以不用{}(大括号)括起来

3、Java中，在如下所示的Test类中，共有（D ）个构造方法。

public class Test{

private int x;

public Test(){

x=35;

}

public void Test(double f){ //构造方法中，不能使用void关键字。

This.x=(int)f;

}

public Test(String s){}

}

0

1

2

3

4、变量命名规范说法正确的是（ B ）

变量由字母、下划线、数字、$符号随意组成；

变量不能以数字作为开头；

A和a在java中是同一个变量；

不同类型的变量，可以起相同的名字；

5、下列javaDoc注释正确的是（ AC ）

/\*我爱北京天安门\*/

//我爱北京天安门\*/

/\*\*我爱北京天安门\*/

/\*我爱北京天安门\*\*/

6、为一个boolean类型变量赋值时，可以使用( B )方式

boolean = 1;

boolean a = (9 >= 10);

boolean a="真";

boolean a = = false;

7、以下( C )不是合法的标识符

STRING

x3x;

void

de$f

8、表达式(11+3\*8)/4%3的值是( D )

31

0

1

2

/9、（ A ）表达式不可以作为循环条件

i++;

i>5;

bEqual = str.equals("q");

count = = i;

/10、运算符优先级别排序正确的是（C ）

由高向低分别是：()、!、算术运算符、关系运算符、逻辑运算符、赋值运算符；

由高向低分别是：()、关系运算符、算术运算符、赋值运算符、!、逻辑运算符；

由高向低分别是：()、算术运算符、逻辑运算符、关系运算符、!、赋值运算符；

由高向低分别是：()、!、关系运算符、赋值运算符、算术运算符、逻辑运算符；

/11、以下程序的运行结果是：( B )

public class Increment{

public static void main(String args[]) {

int a;

a = 6;

System.out.print(a);

System.out.print(a++);

System.out.print(a);

}

}

666

667

677

676

/12、下列值不为true的表达式有（ C ）。

"john" = = "john"

"john".equals("john")

"john" = "john"

"john".equals(new String("john"))

/13、下列输出结果是（ C ）

int a = 0 ;

while ( a < 5 ) {

switch(a){

case 0:

case 3 : a = a + 2;

case 1 :

case 2 : a = a + 3;

default : a = a + 5;

}

}

System.out.print ( a ) ;

0

5

10

其他

/14、下列代码输出结果是( B )

int i = 10;

while ( i > 0 ){

i = i + 1;

if ( i = =10 ){

break;

}

}

while循环执行10次

死循环

循环一次都不执行

循环执行一次

/15、下面有关for循环的描述正确的是（ A ）

for循环体语句中，可以包含多条语句，但要用大括号括起来

for循环只能用于循环次数已经确定的情况

在for循环中，不能使用break语句跳出循环

for循环是先执行循环体语句，后进行条件判断

/16、下列（ BD ）属于引用数据类型（选择两项）

String

Char //基本数据类型

用户自定义的Student类类型

Int //基本数据类型

/17、对象的特征在类中表示为变量，称为类的（ B ）。

对象

属性

方法

数据类型

/18、在java中下列关于自动类型转换说法正确的是（ AB ）

基本数据类型和String相加结果一定是字符串型

char类型和int类型相加结果一定是字符

double类型可以自动转换为int

char + int + double +"" 结果一定是double；

/19、关于类的描叙正确的是（ BD ）（选择两项）

在类中定义的变量称为类的成员变量，在别的类中可以直接使用

局部变量的作用范围仅仅在定义它的方法内，或者是在定义它的控制流块中

使用别的类的方法仅仅需要引用方法的名字即可

一个类的方法使用该类的另一个方法时可以直接引用方法名

/20、下列关于字符串的描叙中错误的是（AC ）（选择两项）

字符串是对象 //String a=””; a是对象而且是字符串，所以字符串是对象

String对象存储字符串的效率比StringBuffer高 //应该更低

可以使用StringBuffer sb="这里是字符串"声明并初始化StringBuffer对象sb

String类提供了许多用来操作字符串的方法：连接，提取，查询等

填空题

1. 如果子类中的某个变量的变量名与它的父类中的某个变量完全一样，则称子类中的这个变量\_\_\_\_\_\_\_\_了父类的同名变量。（隐藏）

2. 属性的隐藏是指子类重新定义从父类继承来的\_\_\_\_\_\_\_\_\_\_。（同名变量或属性）

3. 如果子类中的某个方法的名字、返回值类型和\_\_\_\_\_\_\_\_与它的父类中的某个方法完全一样，则称子类中的这个方法覆盖了父类的同名方法。（参数列表）

4. Java 仅支持类间的\_\_\_\_重继承。（单重）

5. 抽象方法只有方法头，没有\_\_\_\_\_\_\_\_。 （方法体）

解答题

**1.运行下面的程序,结果一般会输出\_\_**  
 public class J\_test{  
    public static void main(String[] args) {  
         try {  
               System.out.println("1");  
               if(true){  
                        return;

               }  
        }catch(RuntimeException e) {  
              System.out.println("2");  
              return;  
       }finally{  
               System.out.println("4");  
       }  
         System.out.println("End!");    
    }  
 }  
**2.运行下面的程序,结果一般会输出\_\_**  
 public class J\_test{  
    public static void main(String[] args) {  
          try {  
                  System.out.println("1");  
                  if(true){  
                          System.exit(0);                      }  
          }catch(RuntimeException e) {  
                   System.out.println("2");  
                   return;  
          }finally{  
                   System.out.println("4");  
          }  
                   System.out.println("End!");    
    }  
 }  
**3.运行下面的程序,结果一般会输出\_\_**  
public class J\_test{  
    public static void main(String[] args) {  
         int i = 0;  
         try {  
                for (i = 0; i < 10; i++) {  
                       try {  
                               if(i%3 == 0){  
                                       throw new Exception();                               }  
                               System.out.println("1:"+1+",");  
                       }catch(Exception e) {  
                               System.out.println("2:"+i+",");  
                               i += 2;  
                               if(i%3 == 2){  
                                       throw new Exception();                               }  
                               return;  
                      }finally{

                                i \*= 2;  
                              System.out.println("3:"+i+",");  
                      }  
                }  
          }catch(Exception e){  
                     System.out.println("4:"+i+",");  
                     return;

          }finally{  
                     System.out.println("5:"+i+",");    
          }  
                     System.out.println("End!");  
       }  
}  
**4.下面程序输出的是\_\_**  
public class J\_test{  
          public static void mb\_method(int i){  
                   try {  
                          if(i ==1){  
                          throw new Exception();  
                          }  
                          System.out.println("1");  
                   }catch(Exception e){  
                          System.out.println("2");  
                          return;  
                   }finally{  
                           System.out.println("3");  
                   }  
                       System.out.println("4");    
           }  
     public static void main(String[] args) {  
               mb\_method(0);

               mb\_method(1);

}  
}

问答题：

1、final,finally,finalize的区别

2、抽象类和接口的区别重写和重载的区别

3、重写toString方法有什么作用

4、== 和equals()在对象比较时有什么不同

5、Java的oob有什么特点