# 面向对象高级

## 一、选择题

1. 有语句String s="hello world"; 以下操作不合法的是( A )。

A. s>>>=1;

B. int i=s.length();

C. String ts=s.trim();

D. String t=s+"!";

1. 请看下面的代码：

String s = "people";

String t = new String( "people");

char c[] ={ 'p', 'e', 'o', 'p', '1', 'e' }

下面选项中语句返回值为真的是( A )

A. s. equals(t);

B. s==t;

C. t.equals(c);

D. t==c;

1. 若有定义String tom="boy"; String hill ="boy";则表达式(ton==hill)的值为( B )。

A. 1 B. true C. false D. 0

1. 编译代码class MyString extends String { }会出现的情况是( D )

A. 成功编译

B. 不能编译，因为没有main方法

C. 不能编译，因为String是abstract类型的

D. 不能编译，因为String是final类型的

1. 以下关于String类说法错误的是( D )

A. 它是定义在 java.lang 包下的一个类

B. 它不是基本数据类型

C. String是不可变的

D. "hello" == new String("hello") 为true

1. 以下不是String类被final修饰的好处的是( D )

A. 由于String是不可变类，所以在多线程中使用是安全的，我们不需要做任何其他同步操作

B. String是不可变的，它的值也不能被改变，所以用来存储数据密码很安全

C. 因为java字符串是不可变的，可以在java运行时节省大量java堆空间。因为不同的字符串变量可以引用池中的相同的字符串。如果字符串是可变得话，任何一个变量的值改变，就会反射到其他变量，那字符串池也就没有任何意义了。

D. 加快字符串的连接操作如s1 + "abc"

1. Java语言中，String类中的indexOf()方法返回值的类型( BC )
2. int16 //相当于short

B. int32 //相当于int

C. int

D. long

1. 执行下列代码后，哪个结论是正确的( B )

String[] s=new String[10];

A. s[10]为””

B. s[9]为null

C. s[0]为未定义

D. s.length为11

1. 有定义：String str; str中存放了英文字母，以下代码不能将str变为在str原有字符上追加“good”字符串的选项是( A )

A. str.concat("good")

B. str = str.concat ("good");

C. str = str + "good";

D. str = new String(str + "good");

1. 以下代码不能将字符串String str = "100"; 转换为对应的整型的是( D )

A. Integer num = Integer.valueOf(str);

B. Integer num = new Integer(str);

C. Integer num = Integer. parseInt(str);

D. Integer num = (Integer) str;

1. 以下代码不能将整型Integer num = 100; 转换为对应的字符串的是( D )

A. String str = num.toString();

B. String str = num + "";

C. String str = String.valueOf(num);

D. String str = (String) num;

1. 以下关于为什么Java中需要包装类的说法不正确的是( D )

A. Java 的设计思想是万物既对象，包装类体现了面向对象的设计理念

B. 包装类包含了很多属性和方法，比基本数据类型功能多

C. 包装类变量可以被赋予null

D. 极大加快了程序的运行速度

1. 以下状况不会发生自动装箱的是( D )

A. 在执行方法调用时，传递一个基本类型的值给到一个方法参数为包装类型的方法中

B. 将一个基本类型的值赋值给一个包装类型对象

C. 方法的返回类型是Integer，返回语句return 10;

D. int num = 10;

14. 以下状况不会发生自动拆箱的是( D )

A. Integer a = 1; boolean b = a > 2;

B. Integer a = 1; Integer b = 2; int c = a + b;

C. Integer a = 1; int b = a;

D. int a = 1 + 2;

## 二、填空题

1. 使用String对象的 toUpperCase 方法将字符串对象的所有字母变为响应的大写字母。
2. 使用String对象的 trim 方法可以将字符串前后的空格去掉。

## 三、简答题

1. 写出以下代码的运行结果

public class Example {

String str=new String("good");

char [] ch={'a','b','c'};

public void change(String str,char ch[]){

str="test ok";

ch[0]='g';

}

public static void main(String[] args) {

Example ex=new Example();

ex.change(ex.str, ex.ch);

System.out.print(ex.str+"and");

System.out.print(ex.ch);

}

}

goodandgbc

str="test ok";执行的结果不会反映到函数以外，char是可以改变的，print（char数组名）会自动打印整个数组，因为PrintStream有多个重载的println方法，其中一个就是public void println(char[] x)，可以直接打印整个字符数组

1. 写出以下代码的运行结果

public class Test {

public static void main(String[] args) {

String s1 = new String("Hello");

String s2 = new String("Hello");

System.out.print(s1 == s2); false

String s3 = "Hello";

String s4 = "Hello";

System.out.print(s3 == s4); true

s1 = s3;

s2 = s4;

System.out.print(s1 == s2); false

}

}

1. 写出以下代码的运行结果

public class Test {

public static void main(String[] args) {

Double num1 = 100D;

Double num2 = 100D;

System.out.print(num1 == num2); false

System.out.print(num1.equals(num2)); true

}

}