**中软国际考题-笔试**

1. 单选题

1、从JDK哪个版本开始Java支持自动装箱和拆箱（B）

A.1.4.3 B.1.5 C.1.6 D.1.7

2、以下哪个是char类型的包装类型（C）

A. Char B.String C.Character D.CharSquence

3、以下代码执行的结果是？（A）

Integer i1 = 100;

Integer i2 = 100;

Integer i3 = 200;

Integer i4 = 200;

System.out.println(i1==i2);

System.out.println(i3==i4);

A．true false B.true true C.false true D.false false

4、下列描述中正确的是（B）

A. 装箱拆箱只限于八大基本类型运行 B. 装箱简单的说就是将值类型转化为包装类类型

C. 装箱拆箱均为隐式操作 D. 拆箱就是把符合容器类转化为值类型的数据

5、下列关于装箱拆箱描述错误的是（B）

A. 数据类型可以规定泛型自动装箱拆箱 B. 装箱拆箱只允许自动进行

C. 基本数据类型不是对象 D. 自动装箱拆箱与对象有关。

6、java中装箱拆箱描述正确的是（D）

A．基本数据类型没有可调用的方法

B. 基本数据类型的自动装箱(autoboxing)、拆箱(unboxing)是自J2SE 4.0开始提供的功能

C. 自动装箱和拆箱从Java 1.6开始引入

D. 自动装箱时编译器调用valueOf将原始类型值转换成对象

7、已知Integer i1 = 300，Integer i2 = 300.问 i1 == i2为true正确吗？( B)

A．正确

B. 错误

8. 自动装箱机制是以下哪个版本开始有的 ( C )

A. JDK 1.1 B. JDK 1.9 C .JDK1.5 D. JDK20

9. 自动拆箱是什么时机完成的 ( A )

A. 编译时自动完成 B.类加载时 C.对象实例化时 D.需要手动完成

10、Number包装器中不包括（B）

A. Integer B. Character C. Long D. Byte

11、请选择以下代码运行结果?（A）

Integer i1=126;

Integer i2=126;

System.out.println("i1==i2"+(i1==i2));

A、true B、false C、编译错误 D、运行错误

12、请选择以下代码运行结果？（B）

Integer io1=new Integer(10);

Integer io2=new Integer(10);

System.out.println("i1==i2"+(i1==i2));

A、true B、false C、编译错误 D、运行错误

13、请选择以下代码运行结果?（B）

Integer i3=1000;

Integer i4=1000;

System.out.println("io3==io4"+(io3==io4));

A、true B、false C、编译错误 D、运行错误

14、请选择以下代码运行结果?（B）

Integer io3=new Integer(1000);

Integer io4=new Integer(1000);

System.out.println("io3==io4"+(io3==io4));

A、true B、false C、编译错误 D、运行错误

15、请选择以下代码运行结果?（A）

Character c1=-13;

Character c2=-13;

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

A、编译出错 B、运行出错C、null D、””

16、请选择以下代码运行结果?（A）

Character c3=13;

Character c4=13;

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

A、true B、false C、编译错误 D、运行错误

17、请选择以下代码运行结果?（A）

Boolean bo1=true;

Boolean bo2=true;

System.out.println("bo1==bo2"+(bo1==bo2));

A、true B、false C、编译错误 D、运行错误

18、请选择以下代码运行结果?（A）

Byte b1=10;

Byte b2=10;

System.out.println("b1==b2"+(b1==b2));

A、true B、false C、编译错误 D、运行错误

19、Java自动装箱和拆箱是从Java那个版本开始引入（ B ）？

A. 1.4 B. 1.5

C. 1.6 D. 1.7

20， Java自动装箱时编译器调用（ C ）方法将原始类型值转换为对象。

A. intValue() B. clone()

C. valueOf() D. toString()

21， 一个整数对象在自动拆箱时， Java编译器会调用（ D ）方法。

A. integerValue() B. valueOf()

C. value() D. intValue()

22、装箱中可能用到的包装类不包括的是（B）

A.Integer B.Char C.Double D.Short

23、下列语句序列执行后打印的结果是（B）

Integer value5 = 128;

Integer value6 = 128;

System.out.println(value5==value6);

System.out.println(value5.equals(value6));

A.false false B. false true C.true true D.true false

24、下列语句序列执行后打印的结果是（B）

Integer value5 = 127;

Integer value6 = 127;

System.out.println(value5==value6);

System.out.println(value5.equals(value6));

A.false false B. false true C.true true D.true false

25、下列语句序列执行后打印的结果是（B）

Double value5 = -128.0;

Double value6 = -128.0;

System.out.println(value5==value6);

System.out.println(value5.equals(value6));

A.false false B. false true C.true true D.true false

26、下列代码的执行结果是（A）

public class One {

public static void go(Long n) {

System.out.println("Long ");

}

public static void go(Short n) {

System.out.println("Short ");

}

public static void go(int n) {

System.out.println("int ");

}

public static void main(String[] args) {

short y = 6;

long z = 7;

go(y);

go(z);

}

}

1. int,Long B. Short,Long C. Long,Long D. Long, Short
2. 多选题
3. 以下哪些类型在执行自动装箱的过程中会使用缓存池优化性能（A、B）

A．int B. short C. double D.float

2、以下哪些属于基本类型的包装类型？(A、C、D)

A. Integer B.Char C.Boolean D.Float

3、以下关于匿名类说法错误的是？（A、C）

A. boolaen值可以转换成Boolaen对象 B. char可以转换成Character对象

C. int可以转换成Intager D. byte值转换成Byte对象

4、下列答案中哪个是正确的？(A C )

public class Main {

public static void main(String[] args) {

Double i1 = 100.0;

Double i2 = 100.0;

Double i3 = 200.0;

Double i4 = 200.0;

System.out.println(i1==i2);

System.out.println(i3==i4);

}

}

    A．i1==i2 为false

    B．i1==i2 为true

    C．i3==i4 为false

D．i3==i4 为true

5、包装器的特性？（A、B）

A. 包装器是不可变的，一旦构造了包装器，就不允许更改包装在其中的值

B. 包装器是final定义的，不允许定义它的子类

C. 所有类型都有包装器

D. 包装器的valueOf方法有相同的实现和相同的范围

6、以下哪些是包装器类型？（BC）

A、Int B、Character C、Boolean D、true

7、下面描述中正确的是？（B、C、D）

A. Integer i=10; 类型不匹配会报错

B. Integer i=10.5; 类型不匹配会报错

C、Integer i = 100;相当于编译器自动作以下的语法编译：Integer i = Integer.valueOf(100);

D. Integer i = 10; int t = i; //拆箱，实际上执行了 int t = i.intValue();

8、下面哪些项结果为True？（A、B）

A. Integer i1 = -128; Integer i2 = -128; System.out.println(i1==i2);

B. Integer i3 = 127; Integer i4 = 127; System.out.println(i3==i4);

C. Integer i5 = -129; Integer i6 = -129; System.out.println(i5==i6);

D. Integer i7 = 128; Integer i8 = 128; System.out.println(i7==i8);

9、下面运行结果为true的选型为？（A、C、D、E）

public class AutoBoxing{

public static void main(String[] args){

Integer a = 1;

Integer b = 2;

Integer c = 3;

Integer d = 3;

Integer e = 321;

Integer f = 321;

Long g = 3L;

A. System.out.println(c==d); B. System.out.println(e==f);

C. System.out.println(c==(a+b)); D. System.out.println(c.equals(a+b));

E. System.out.println(g==(a+b)); F. System.out.println(g.equals(a+b));

1. 填空题（每空1分，共10分）
2. 在针对long型数据进行自动装箱时，触发缓存池性能优化的最小值和最大值分别是\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(-128、127)

2、针对数字类型进行自动装箱拆箱是使用的缓存池以\_\_\_\_\_\_\_\_\_\_形式存在。（静态的Integer数组）

3、原始类型byte,short,char,int,long,float,double和boolean对应的封装类为

\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_、\_\_\_\_\_\_\_\_\_。

(Byte,Short,Character,Integer,Long,Float,Double,Boolean)

4、Java JDK1.5之后有自动装箱和自动拆箱

5、Java装箱过程是通过调用包装器的\_\_\_\_\_\_\_\_方法实现的，而拆箱过程是通过调用包装器的\_\_\_\_\_\_\_\_方法实现的。(valueOf、xxxValue)

6、包装器类型都包含哪些？：\_\_\_\_\_\_\_。(8种基本数据类型对应着一个包装器类)

7、String类型如何转换为基本数据类型？\_\_\_\_\_\_\_。（调用包装类的parseXxx(String str)方法）

8、boolean类型对应的包装类型是 。 （ Double ）

9、char类型对应的包装类型是 。（ Character ）

10、int类型对应的包装类型是 。（ Integer ）

11、 float类型对应的包装类型是 。（ Float ）

12、Integer value5 = 128;  
Integer value6 = 128;  
System.out.println(value5==value6); //false  
System.out.println(value5.equals(value6)); //true

程序的执行结果是\_\_\_\_\_\_\_\_\_\_ 、\_\_\_\_\_\_\_\_\_。(false true)

13、请分别写出byte、char、int类型的包装类：\_\_\_\_Byte\_\_\_\_\_\_、\_\_\_\_Character\_\_\_\_\_\_ 、\_\_\_Integer\_\_\_\_\_\_。

14、将一个原始数据类型赋值给相应封装类的变量，叫做\_\_\_\_\_\_\_\_\_\_ , 将一个Integer类型的对象赋值给其相应原始数据类型的变量，叫做\_\_\_\_\_\_\_\_\_。(装箱、拆箱)

1. 判断题
2. 对于Integer，在(-128,128]之间只有固定的256个值，所以为了避免多次创建对象，事先创建好一个大小为256的Integer数组SMALL\_VALUES，如果值在这个范围内，就可以直接返回事先创建好的对象就可以了，但是对于Double类型来说，就不能这样做，因为它在这个范围内个数是无限的。（ T ）
3. 代码int[] a={1};Integer[] b = a;的写法体现了Java中的自动装箱机制。（F）
4. 引入了具有自动转换的装箱和拆箱机制是为了jdk的升级。（ N ）
5. **Integer i = 120（ T ）**
6. **Interger i = 20,不能赋值给整型变量( F )**
7. Java装箱就是自动将基本数据类型转换为包装器类型。（ T ）
8. Java拆箱就是自动将包装器类型转换为基本数据类型（ T ）
9. Double i1 = 100.0; Double i1 = 100.0; i1==i2 结果是true（F）
10. 赋值语句 Integer i3 = 12;是正确的（ T ）
11. 原始值的比较使用 == 符号。 （ T ）
12. 使用 == 符号比较对象时， 比较的是对象的值。 （ F ）
13. Java自动将原始类型值转换成对应的对象，比如将int的变量转换成Integer对象，这个过程叫做装箱。（ T ）
14. 基本数据类型的自动装箱(autoboxing)、拆箱(unboxing)是自J2SE 6.0开始提供的功能。（F）
15. Intger 包装类型的常量池为 0 到 127 之间（ F ）
16. Integer i1 = 127，Integer i2 = 127, i1==i2的结果为true。（ T ）
17. Integer a = new Integer(127), Integer b = 127,a==b的结果为true。（ F ）

五、简答题

1、以下代码的执行结果是什么？

public class Main {

public static void main(String[] args) {

Integer a = 1;

Integer b = 2;

Integer c = 3;

Integer d = 3;

Integer e = 321;

Integer f = 321;

Long g = 3L;

Long h = 2L;

System.out.println(c==d);

System.out.println(e==f);

System.out.println(c==(a+b));

System.out.println(c.equals(a+b));

System.out.println(g==(a+b));

System.out.println(g.equals(a+b));

System.out.println(g.equals(a+h));

}

}

答案：

true

false

true

true

true

false

true

2、请简述自动装箱拆箱要点（5分）

答案： 自动装箱时编译器调用valueOf将原始类型值转换成对象，同时自动拆箱时，编译器通过调用类似intValue(),doubleValue()这类的方法将对象转换成原始类型值。

自动装箱是将boolean值转换成Boolean对象，byte值转换成Byte对象，char转换成Character对象，float值转换成Float对象，int转换成Integer，long转换成Long，short转换成Short，自动拆箱则是相反的操作。

3、java1.5的自动装箱拆箱机制是编译特性还是虚拟机运行时特性？分别是怎么实现的？

答案：java1.5开始的自动装箱拆箱机制其实是编译时自动完成替换的，装箱阶段自动替换为了valueOf方法，拆箱阶段自动替换为了xxxValue方法。

对于Integer、Short、Byte、Character、Long类型的valueOf方法，参数如果是-128~127之间的值会直接返回内部缓存池中已经存在对象的引用，参数是其他范围值则返回新建对象；

而Double、Float类型与Integer类型类似，一样会调用Double、Float的valueOf方法，但是不管传入的参数值是多少都会new一个对象来表达该数值，因为在某个范围内的整型数值的个数是有限的，而浮点数却不是。

4、简述Java的包装器类型有哪些？

答案： Short、 Integer、Long、Character、 Byte、 Float、 Boolean、Double

5、什么是装箱？什么是拆箱？

答案：装箱就是自动将基本数据类型转换为包装器类型；拆箱就是自动将包装器类型转换为基本数据类型。

6、Java中为什么需要自动装箱与拆箱？

答： 因为在Java中， 你不能直接地向集合(Collections)中放入原始类型值，集合只接收对象。通常这种情况下你的做法是，将这些原始类型的值转换成对象，然后将这些转换的对象放入集合中。为了让代码简练，Java 1.5 引入了具有在原始类型和对象类型自动转换的装箱和拆箱机制。

7， 写出Java中原始类型对应的包装类型。

答， 原始类型byte, short, char, int, long, float, double 和 boolean对应的封装类为Byte, Short, Character, Integer, Long, Float, Double, Boolean。

8， Java中， 什么是自动装箱和拆箱？

答： 自动装箱就是Java自动将原始类型值转换成对应的对象，比如将int的变量转换成Integer对象，这个过程叫做装箱，反之将Integer对象转换成int类型值，这个过程叫做拆箱。

9, Java中，何时发生自动装箱和拆箱？用代码简单的演示。

答： 自动装箱和拆箱在Java中很常见，比如我们有一个方法，接受一个对象类型的参数，如果我们传递一个原始类型值，那么Java会自动讲这个原始类型值转换成与之对应的对象。

比如向ArrayList这样的容器中增加原始类型数据时：

ArrayList<Integer> intList = new ArrayList<Integer>();

intList.add(1); //autoboxing - primitive to object

intList.add(2); //autoboxing

int number = intList.get(0); // unboxing

10， 举出例子说明自动装箱的弊端

答：自动装箱有一个问题，那就是在一个循环中进行自动装箱操作的情况，如下面的例子就会创建多余的对象，影响程序的性能。

Integer sum = 0;

for(int i=1000; i<5000; i++){

sum+=i;

}

上面的代码sum+=i可以看成sum = sum + i，但是+这个操作符不适用于Integer对象，首先sum进行自动拆箱操作，进行数值相加操作，最后发生自动装箱操作转换成Integer对象。其内部变化如下

sum = sum.intValue() + i;

Integer sum = new Integer(result);

由于我们这里声明的sum为Integer类型，在上面的循环中会创建将近4000个无用的Integer对象，在这样庞大的循环中，会降低程序的性能并且加重了垃圾回收的工作量。因此在我们编程时，需要注意到这一点，正确地声明变量类型，避免因为自动装箱引起的性能问题。

11，在Java对象比较时， 应该注意哪些要点？

答： 判断两个对象是否为同一个对象， 使用符号==， 这个过程中没有发送自动装箱。比较两个对象的值是否相同， 使用方法equals。 如下的代码：

比如下面的代码：

int i1 = 1;

int i2 = 1;

System.out.println("i1==i2 : " + (i1 == i2)); // true

Integer num1 = 1; // autoboxing

int num2 = 1;

System.out.println("num1 == num2 : " + (num1 == num2)); // true

Integer obj1 = 1; // autoboxing will call Integer.valueOf()

Integer obj2 = 1; // same call to Integer.valueOf() will return same cached Object

System.out.println("obj1 == obj2 : " + (obj1 == obj2)); // true

上面的例子中， obj1和obj2的初始化都发生了自动装箱操作。但是处于节省内存的考虑，JVM会缓存-128到127的Integer对象。因为obj1和obj2实际上是同一个对象。所以使用”==“比较返回true。

Integer one = new Integer(1); // no autoboxing

Integer anotherOne = new Integer(1);

System.out.println("one == anotherOne : " + (one == anotherOne)); // false

12, 在编写循环代码时， 需要注意哪些的问题？

答： 因为自动装箱会隐式地创建对象，像前面提到的那样，如果在一个循环体中，会创建无用的中间对象，这样会增加GC压力，拉低程序的性能。所以在写循环时一定要注意代码，避免引入不必要的自动装箱操作。

13、谈谈Integer i = new Integer(xxx)和Integer i =xxx;这两种方式的区别。

答案：

　　1）第一种方式不会触发自动装箱的过程；而第二种方式会触发；

2）在执行效率和资源占用上的区别。第二种方式的执行效率和资源占用在一般性情况下要优于第一种情况（注意这并不是绝对的）。

14、简述什么是自动装箱和自动拆箱。

答案：1）自动装箱：Java自动将原始类型值转换成对应的对象，比如将int的变量转换成Integer对象，这个过程叫做装箱。

2）自动拆箱就是：反之将Integer对象转换成int类型值，这个过程叫做拆箱。因为这里的装箱和拆箱是自动进行的非人为转换，所以就称作为自动装箱和拆箱。

15、问：java 是否存在使得语句 i > j || i <= j 结果为 false 的 i、j 值？

答案：答：存在，java 的数值 NaN 代表 not a number，无法用于比较，例如使 i = Double.NaN; j = i; 最后 i == j 的结果依旧为 false，这是一道非常变态的题，巨坑，谁特么会这么用。

16、请列举基本数据类型及其对应的包装类。（5分）

答案：byte Byte

short Short

int Integer

long Long

float Float

double Double

char Character

boolean Boolean

17、为什么会有基本类型的包装类？

答案：将基本类型数据类型封装成对象，这样的好处可以在对象中定义更多方法操作该数据，也可以更好满足Java语言面向对象的这一特性。

18、包装类的作用是什么？

答案：1.集合不允许存放基本数据类型，故常用包装类

2.包含了每种基本数据类型的相关属性，如最大值、最小值、所占位数等

3.作为基本数据类型对应的类类型，提供了一系列实用的对象操作，如类型转换、进制转换等等

19、包装类中”==”和equals有什么区别？

答案：包装类中的equals方法和String类一样，都是重写了Object类中的equals方法，因此比较的是内容而不是地址，而“==”比较的依然是引用变量的地址，只是当包装类型和与之相对应的基本类型进行“==”比较时会先做自动拆箱处理。

六、程序实现题

1、根据自动装箱过程中的缓存池操作机制，编写针对int数进行装箱的缓存IntegerCache实现和Integer的valueOf方法

|  |
| --- |
| class IntegerCache {  static final int high;  static final Integer cache[];  static {  final int low = -128;  // high value may be configured by property  int h = 127;  if (integerCacheHighPropValue != null) {  // Use Long.decode here to avoid invoking methods that  // require Integer's autoboxing cache to be initialized  int i = Long.decode(integerCacheHighPropValue).intValue();  i = Math.max(i, 127);  // Maximum array size is Integer.MAX\_VALUE  h = Math.min(i, Integer.MAX\_VALUE - -low);  }  high = h;  cache = new Integer[(high - low) + 1];  int j = low;  for(int k = 0; k < cache.length; k++)  cache[k] = new Integer(j++);  }  private IntegerCache() {}  }  public static Integer valueOf(int i) {  if(i >= -128 && i <= IntegerCache.high)  return IntegerCache.cache[i + 128];  else  return new Integer(i);  } |

2、自动装箱拆箱： 实现Java中的自动装箱拆箱（利用泛型机制）（10分）

|  |
| --- |
| public class Test01 {  public static void main(String[] args) {  Map<Integer, Boolean> map=new HashMap<Integer, Boolean>();  map.put( 0, true);  map.put(12, false);  map.put(18, true);  System.out.println(map);  boolean index=map.get(18);  System.out.println(index);  }  } |

3、写出下面程序代码的输出结果

public static void main(String[] args) {

//1

Integer a=new Integer(123);

Integer b=new Integer(123);

System.out.println(a==b);//输出 false

//2

Integer c=123;

Integer d=123;

System.out.println(c==d);//输出 true

//3

Integer e=129;

Integer f=129;

System.out.println(e==f);//输出 false

//4

int g=59;

Integer h=new Integer(59);

System.out.println(g==h);//输出 true

}

4、获取两个字符串的最大相同子串，例如abegad与acegab，最大相同子串为ega。（10分）

|  |
| --- |
| public class MaxSubString {  public String maxSubString(String str1,String str2){  int len1=str1.length();  int len2=str2.length();  char[] strc1=str1.toCharArray();  char[] strc2=str2.toCharArray();  int[][] map=new int[len1][len2];  System.out.print(" ");  for(int j=0;j<len2;j++){  System.out.print(strc2[j]+" ");  }  System.out.println();  for(int i=0;i<len1;i++){  System.out.print(strc1[i]+" ");  for(int j=0;j<len2;j++){  if(strc1[i]==strc2[j]){  if(i==0||j==0){  map[i][j]=1;  }  else{  map[i][j]=map[i-1][j-1]+1;  }    }  System.out.print(map[i][j]+" ");  }  System.out.println();    }  System.out.println();  System.out.println();  System.out.println();  int maxi=0;  int maxj=0;  for(int i=0;i<len1;i++){  for(int j=0;j<len2;j++){  if(map[i][j]>map[maxi][maxj]){  maxi=i;  maxj=j;  }  }  }  if(map[maxi][maxj]!=0){  StringBuffer max=new StringBuffer("");  while(maxi>=0&&maxj>=0&&map[maxi][maxj]!=0){  max.insert(0, strc1[maxi]);  maxi--;  maxj--;  }  return max.toString();  }else{  return "";  }  }  public static void main(String s[]){  MaxSubString max=new MaxSubString();  System.out.println(max.maxSubString("abc1234-122d", "kjadjasdc1234-122ab"));  System.out.println(max.maxSubString("wex\_22y8jhrevx", "we45x\_22yt08jhrevx"));  }  } |

5、以下代码的输出结果是？

（10分）

|  |
| --- |
| public class Main {      public static void main(String[] args) {          Integer a = 1;          Integer b = 2;          Integer c = 3;          Integer d = 3;          Integer e = 321;          Integer f = 321;          Long g = 3L;          Long h = 2L;          System.out.println(c==d);          System.out.println(e==f);          System.out.println(c==(a+b));          System.out.println(c.equals(a+b));          System.out.println(g==(a+b));          System.out.println(g.equals(a+b));          System.out.println(g.equals(a+h));      } |

true

false

true

true

true

false

true

6、Integer i1 = 100;

Integer i2 = 100;

Integer i3 = 200;

Integer i4 = 200;

System.out.println(i1==i2);//true

System.out.println(i3==i4);//false

Double i1 = 100.0;

Double i2 = 100.0;

Double i3 = 200.0;

Double i4 = 200.0;

System.out.println(i1==i2);//false

System.out.println(i3==i4);//false

Boolean i1 = false;

Boolean i2 = false;

Boolean i3 = true;

Boolean i4 = true;

System.out.println(i1==i2);//true

System.out.println(i3==i4);//true

Integer a = 1;

Integer b = 2;

Integer c = 3;

Integer d = 3;

Integer e = 321;

Integer f = 321;

Long g = 3L;

Long h = 2L;

System.out.println(c==d);//true 比较cache

System.out.println(e==f);//false 比较引用

System.out.println(c==(a+b));//true 比较值

System.out.println(c.equals(a+b));//true 比较值

System.out.println(g==(a+b));//true //比较值

System.out.println(g.equals(a+b));//false //比较引用(对于包装器类型，equals方法并不会进行类型转换)

System.out.println(g.equals(a+h));//true比较引用

Integer a = 444;

int b = 444;

System.out.println(a==b); //true 比较值

System.out.println(a.equals(b));//true比较值

对于Integer、Short、Byte、Character、Long类型的valueOf方法，参数如果是-128~127之间的值会直接返回内部缓存池中已经存在对象的引用，参数是其他范围值则返回新建对象；

而Double、Float类型与Integer类型类似，一样会调用Double、Float的valueOf方法，但是不管传入的参数值是多少都会new一个对象来表达该数值，因为在某个范围内的整型数值的个数是有限的，而浮点数却不是。

7、将字符串s = “姓名：赵丽颖，年龄，20”中的20取出来，并计算乘以5的结果。

（10分）

|  |
| --- |
| **public** **class** Test3 {  **public** **static** **void** main(String[] args) {  String s = "姓名:赵丽颖,年龄:20";  String ageS = s.split(",")[1].split(":")[1];  **int** age = Integer.*parseInt*(ageS);  System.***out***.println(age\*5);  }  } |