

# Java 自测试题

注意：

1. 下面的题目不包含JDBC和网络编程，后续我们会有大作业。

2. 题目中只要不要求编程，不要实际去运行程序。

3. 除非特别注明，每道题均是1分

4. 请自测得分，然后把结果输入到

<https://shimo.im/spreadsheet/DGZftdwmBCUpf4mg>

注意，不要修改别人的得分，以免混乱

## 第一部分：Java 概述

1.1 Java 是怎么做到跨平台的？

1.2 Java EE, Java SE, Java ME分别是什么？

1.3 简述JRE与JDK的区别？

1.4 为什么要设置Java Home 和 Path ？

1.5 在运行java 程序时，为什么要设置classpath ？

1.6 一个.java源文件是否可以包含多个类？有什么限制

## 第二部分：基本语法

2.1 以下数据类型分别占多少字节？

byte  
short  
int  
long  
float  
double

2.2 下面语句有什么错误？

```
short s1 = 1;  
s1 = s1 + 1;
```

2.3 int和Integer都可以代表一个整数，为什么JDK设计了这两个类型？

2.4 在脑子里运行下面的代码，x的值最终是多少？

```
public class JavaTest {  
    public static void main(String args[]) {  
        int x=2 ;  
        do {  
            x+=x ;  
        }  
        while(x<17);  
        System.out.println(x);  
    }  
}
```

2.5 请看下面的程序代码：

```
switch(n){  
    case 0: System.out.println("first");  
    case 1:
```

```

        case 2: System.out.println("second"); break;
        default: System.out.println("end");
    }

```

当n为何值时，程序段可以输出字符串"second"

- A. 0    B. 1    C. 2    D. 以上都可以

**2.6 请问 下面的代码执行完毕以后， b 的值， x的值， y的值各是多少？**

```

int x=6,y=8;
boolean b;
b=x>y || ++x==--y;

```

**2.7 请问 下面的代码执行完毕以后， b 的值， x的值， y的值各是多少？**

```

int x=6,y=8;
boolean b;
b=x>y && ++x==--y;

```

**2.8 下面的程序输出是什么？**

```

public class JavaTest {

    public static void changeStr(String str){
        str="welcome";
    }
    public static void main(String[] args) {

        String str="1234";
        changeStr(str);
        System.out.println(str);
    }
}

```

**2.9 有代码如下， 请问调用changeValue以后， 数组arr的值是多少？**

```

public class JavaTest {
    private static void changeValue(int[] arr) {
        for (int i = 0; i < arr.length; i++)
            arr[i] *= 2;
    }
    public static void main(String[] args) {
        int[] arr = { 1, 2, 3, 4, 5 };
        changeValue(arr);

    }
}

```

**2.10 下面代码有什么错误？**

```

public class JavaTest {
    final int i;
    public void doSomething() {
        System.out.println("i = " + i);
    }
}

```

**2.11 下面的代码试图计算 2.00 - 1.10, 但是输出的值确是0.8999999999999999 ,这是为什么？**

```

public class JavaTest {

```

```

public static void main(String[] args) {
    System.out.println(2.00-1.10);
}
}

```

### 第三部分：面向对象

#### 3.1 下面代码有什么错误？

```

public class JavaTest {
    public static void main(String[] args) {
        JavaTest s = new JavaTest();
        System.out.println("s.doJavaTest() returns " + doJavaTest());
    }
    public String doJavaTest() {
        return "Do JavaTest ...";
    }
}

```

#### 3.2 下面的代码有什么问题？为什么？

```

public class JavaTest {
    void doSomething() {
        private String s = "";
        int l = s.length();
    }
}

```

#### 3.3 试完成下述程序片段

```

public class Point{
    int x,y;
    public Point(int x,int y){
        ( )=x;
        ( )=y;
    }
    ...
}

```

- A . Point.x Point.y    B . this.x this.y  
C . super.x super.y    D . 无解

#### 3.4 下面的代码有什么问题？

```

public class JavaTest {
    public static void main(String[] args) {
        Hello obj = new Hello();
        obj.msg += ",World!";
        System.out.println(obj.msg);
    }
}

```

```

class Hello {
    public String msg = "Hello";
    public Hello(String msg) {
        this.msg = msg;
    }
}

```

```
}
```

### 3.5 下面的代码有什么问题？为什么？

```
public class JavaTest {  
    public int addOne(final int x) {  
        return ++x;  
    }  
}
```

### 3.6 下面的代码有什么问题？为什么？

```
public class JavaTest {  
    public static void main(String[] args) {  
        Other o = new Other();  
        new JavaTest().addOne(o);  
    }  
    public void addOne(final Other o) {  
        o.i++;  
    }  
}  
class Other {  
    public int i;  
}
```

### 3.7 下面的程序输出什么值？

```
public class JavaTest {  
    public static void main(String args[]) {  
        JavaTest obj = new JavaTest();  
        obj.method(100);  
    }  
    void method(int i) {  
        System.out.println("int: " + i);  
    }  
    void method(long i) {  
        System.out.println("long: " + i);  
    }  
}
```

### 3.8 下面的代码输出什么：

```
class Fruit {  
    String name = "Fruit";  
    public void print(int i) {  
        System.out.println("Fruit" + i);  
    }  
}
```

```
class Apple extends Fruit {  
    String name = "Apple";  
    public void print(int i) {  
        System.out.println("Apple" + i);  
    }  
}
```

```
public class JavaTest {  
    public static void main(String args[]) {  
        Apple apple = new Apple();  
        apple.print(100);  
        System.out.println(apple.name);  
  
        Fruit fruit = apple;  
        fruit.print(100);  
        System.out.println(fruit.name);  
    }  
}
```

```
}
```

### 3.9 有class A定义如下

```
class A {  
    protected int method1(int a, int b) {  
        return 0;  
    }  
}
```

如果有类class B extends A , 在class B中 , 下列哪两个方法定义是有效的 ?

- A. public int method1(int a, int b) { return 0; }
- B. private int method1(int a, int b) { return 0; }
- C. private int method1(int a, long b) { return 0; }
- D. public short method1(int a, int b) { return 0; }
- E. static protected int method1(int a, int b) { return 0; }

### 3.10 有一个类OuterClass定义如下 :

```
public class OuterClass {  
    public void someOuterMethod() {  
        // Line 3  
    }  
}
```

```
public class InnerClass{  
}
```

```
public static void main(String[] argv) {  
    OuterClass o = new OuterClass();  
    // Line 11  
}
```

下面哪个语句能够实例化InnerClass?

- A. new InnerClass(); // At line 3
- B. new InnerClass(); // At line 11
- C. new o.InnerClass(); // At line 11
- D. new OuterClass.InnerClass(); // At line 11

### 3.11 下面的代码输出什么 ?

```
abstract class A{  
    public abstract void before();  
    public abstract void after();  
    public void print(int i){  
        before();  
        System.out.println("A"+i);  
        after();  
    }  
}
```

```
class B extends A{  
    public void before(){  
        System.out.println("before B");  
    }  
    public void after(){  
        System.out.println("after B");  
    }  
}
```

```

public class JavaTest {
    public static void main(String args[]){
        A a = new B();
        a.print(100);
    }
}

```

### 3.12 下面的代码输出结果是什么？

```

class Fruit {
    public void setDate(Object d) {
        System.out.println("Fruit.setDate(Object d)");
    }
}
class Apple extends Fruit {
    public void setDate(Date d) {
        System.out.println("Apple.setDate(Date d)");
    }
}
public class JavaTest3 {
    public static void main(String[] args) {
        Fruit f = new Apple();
        f.setDate(new Date());
    }
}

```

### 3.13 下列关于类的继承的描述，正确的有哪些？

- A. 一个类可以同时继承多个父类
- B. 一个类可以具有多个子类
- C. 子类会自动拥有父类所有的方法
- D. 一个类继承另一个类需要使用 extends 关键字

### 3.14 有java 类如下：

```

public class JavaTest {
    public static void main(String args[]) {
        System.out.println(new JavaTest());
    }
}

```

运行该Java 类，出现的结果是：[JavaTest@15db9742](#)  
为什么会这样？

### 3.15 在面向对象程序中，实现代码复用主要有两种方式：继承和组合，请描述下这两种方式的区别。

## 第四部分：Java 集合

### 4.1 Java 集合框架的Map, Set和List 有什么区别？

### 4.2 为什么Map 接口不继承Collection接口？

### 4.3 为什么要有Iterator?

### 4.4 hashCode() 和 equals()方法是如何被HashMap所使用的？

### 4.5 Map接口提供了哪些集合视图可供使用？

### 4.6 如何决定选用HashMap还是TreeMap？

4.7 我们如何从给定集合那里创建一个synchronized的集合？

4.8 当一个集合被作为参数传递给一个函数时，如何才可以确保函数不能修改它？

4.9 下列关于ArrayList、Vector和LinkedList集合的说法正确是有哪些？

- A. ArrayList集合底层是数组实现，该集合线程不安全
- B. Vector 集合元素的存放是无序的
- C. LinkedList集合底层是链表结构，适合做元素的增删操作
- D. 这三个集合都是List接口的实现类

4.10 下面程序的输出结果是什么？

```
Set<String> set= new HashSet<String>();  
set.add("aaa");  
set.add("bbb");  
set.add("aaa");  
System.out.println(set.size());
```

- A. 编译不通过
- B. 运行时出错
- C. 输出3
- D. 输出2

4.11 下列哪些集合属于Collection体系的子类？

- A. TreeMap
- B. ArrayList
- C. Hashtable
- D. HashSet

## 第五部分 Java 异常

5.1 Error和Exception 有什么区别？

5.2 Java中的检查型异常和非检查型异常有什么区别？

5.3 finally的作用是什么？什么时候会被执行？

5.4 finally, finalize有什么区别？

下面有关Java异常处理的说法错误的是

- A. 一个try块只能有一条catch语句
- B. 一个try块中可以不使用catch语句
- C. catch块不能单独使用，必须始终与try块在一起
- D. finally块不能单独使用，必须始终与try块在一起

5.5 下面代码有什么问题？

```
public static void start() throws IOException, RuntimeException{  
    throw new RuntimeException("Not able to Start");  
}
```

```
public static void main(String args[]) {  
    try {  
        start();  
    }
```

```

} catch (Exception ex) {
    ex.printStackTrace();
} catch (RuntimeException re) {
    re.printStackTrace();
}
}
}

```

### 5.6 把下面4种情况和后面的四种错误做一个匹配

1. `int[] a; a[0] = 0;`
2. JVM正在运行你的程序的时候，突然找不到一个系统相关的类（位于`rt.jar/classes.zip`中）
3. 一个程序从IO Stream中读取数据，并且到达了Stream的结束处
4. 一个程序从一个已经关闭了的IO Stream中读取数据

- a. 出现Error
- b. 编译错误
- c. 抛出检查型异常
- d. 没有错误

### 5.7 下面程序有何错误，该如何修改？

```

public static void cat(File file) {
    RandomAccessFile input = null;
    String line = null;

    try {
        input = new RandomAccessFile(file, "r");
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        return;
    } finally {
        if (input != null) {
            input.close();
        }
    }
}

```

## 第六部分 Java IO

### 6.1 字节流与字符流有什么区别？

### 6.2 下面程序的运行结果是什么？

```

FileOutputStream fos = new FileOutputStream( "c:\\demo.txt" );
fos.write( "abc" );
fos.close();

```

- A. 在C盘创建文件demo.txt,但文件是空的
- B. 在C盘创建文件demo.txt,并写入数据abc
- C. 将C盘已有的文件demo.txt中追加写入abc
- D. 编译失败

### 6.3 下面代码的作用是什么？

```

BufferedReader br = new BufferedReader(new FileReader("c:\\a.txt"));
BufferedWriter bw = new BufferedWriter(new FileWriter("d:\\b.txt"));
String line = null;
while ((line = br.readLine()) != null) {
    bw.write(line);
    bw.newLine();
    bw.flush();
}

```



```
bw.close();  
br.close();
```

- A. 把c盘目录下的a.txt文件内容复制到d盘目录下的b.txt
- B. 把d盘目录下的b.txt文件内容复制到c盘目录下的a.txt
- C. 读取c盘目录下a.txt文件，输出在控制台
- D. 把控制台的内容写入到d盘目录下的b.txt文件中

6.4 (本题7分) 有一个二进制文件shoppingcart.data, 其中存储的数据依次是：

产品价格(double类型)，产品数量(int类型)，产品描述(UTF字符串)

这样的数据可能有多组，写一个程序，读取该二进制文件的内容，计算出该购物车的总价格(即每一项 产品价格\*产品数量 的和)

? shoppingcart.data  
2017/02/03 17:08, 131B

提示：使用DataInputStream

6.5 (本题7分) 写一个程序，读取一个.class文件的前4个字节，转换成十六进制字符，检查是不是：CAFEBABE

## 第七部分 Java 线程

7.1 为什么Java中有多线程编程，而很少提到Java多进程编程？

7.2 创建一个线程有那些方法？

7.3 为什么我们调用start()方法时会执行run()方法，为什么我们不能直接调用run()方法？

7.4 Java 中多线程之间在什么情况下需要同步？

7.5 什么是不可变对象，它对写并发应用有什么帮助？

7.6 现在有t1、t2、t3三个线程，你怎样保证t2在t1执行完后执行，t3在t2执行完后执行？

7.7 有t1, t2, t3 三个线程，都要对 r1, r2, r3, r4 这4个资源进行读写，对每个资源读写之前需要获得锁才能操作，由于线程的乱序执行，可能导致死锁。

请设计一种策略来避免死锁的产生。

## 第八部分 Java 反射

8.1 为什么需要反射？

8.2 获取Class类型的对象有哪些：

- A. Object类的getClass()
- B. class静态属性
- C. 自己创建Class对象
- D. Class类的forName()静态方法

8.3 (本题7分) 有Java 类如下：

```
public class Employee {  
  
    private String id;  
    private String name;  
    private int age;  
    private Employee(){  
    }  
    public Employee(String name, int age) {  
        this.id = "1001";  
        this.name = name;  
        this.age = age;  
    }  
    private String getID(){
```

```

return this.id;
}
public void sayHello(){
    System.out.println("Hello, name =" + name + " age = " + age);
}
}

```

要求：

(1) 通过反射的方式创建Employee的实例

(2) 通过反射调用sayHello()方法

(3) 通过反射调用getId()方法

(4) 打印出每个字段，格式为

权限描述符(private ,protected 等) + 字段类型+字段名称

#### 8.4 有一个支持Integer 的ArrayList如下：

```
ArrayList<Integer> list = new ArrayList<Integer>();
```

通过反射向该ArrayList中加入一个String类型的元素

#### 8.5 (本题8分) 有AccountService, AccountServiceImpl, Transaction 类如下：

```

public interface AccountService {
    public void transfer(String fromAccount,
        String toAccount,
        int ammount);
    public void query(String accountId);
}

public class AccountServiceImpl implements AccountService {

    @Override
    public void transfer(String fromAccount, String toAccount, int ammount) {
        System.out.println("Transfer " + ammount + " from " + fromAccount + " to " + toAccount);
    }

    @Override
    public void query(String accountId){
        System.out.println("query account id:" + accountId);
    }
}

public class Transaction {
    public void beginTx(){
        System.out.println("Begin transaction");
    }

    public void commitTx(){
        System.out.println("commit transaction");
    }
}

```

试用Java 动态代理实现如下功能：

在调用AccountServiceImpl 的transfer("A100", "B200", 233) 方法时，输出如下：

Begin transaction

Transfer 233 from A100 to B200

commit transaction

在调用AccountServiceImpl 的query("A100")，输出结果：

query account id:A100

## 第九部分 Java 泛型

### 9.1 下列说法错误的是

- A. Java中的泛型基本上都是在编译器这个层次来实现的。
- B. 在生成的Java字节码中是不包含泛型中的类型信息的。
- C. 泛型可以应用在接口，类，和方法上。
- D. ArrayList<int> list = new ArrayList<int> 是合法的

### 9.2 下面的代码是否正确？

```
public class AnimalHouse<E> {  
    private E animal;  
    AnimalHouse(){  
        animal = new E();  
    }  
}
```

### 9.3 下面的代码输出的结果是什么？

```
ArrayList<String> list1 = new ArrayList<String>();  
ArrayList<Integer> list2 = new ArrayList<Integer>();  
System.out.println(list1.getClass().equals(list2.getClass()));
```

### 9.4 下面的代码有问题吗？为什么？

```
ArrayList<Number> numbers = new ArrayList<Number>();  
numbers.add(new Integer(10));  
numbers.add(new Double(10.0d));
```

### 9.5 下面的泛型类经过类型擦除以后会变成什么？

```
public class AnimalHouse<T> {  
    private T animal;  
    public T getAnimal(){  
        return this.animal;  
    }  
}
```

### 9.6 泛型类同上一题，有如下代码：

```
AnimalHouse<Tiger> house = new AnimalHouse<Tiger>();  
Tiger t = house.getAnimal();  
经过类型擦除，在执行getAnimal()是JVM会怎么做？
```

### 9.7 下面的代码有什么问题？该如何改正：

```
public static <T> T compare(T t1,T t2){  
    if(t1.compareTo(t2) >= 0){
```

```
    return t1;
} else{
    return t2;
}
}
```

#### 9.8 下面代码的问题是什么？该如何改正？

```
public static void main(String[] args) {
    ArrayList<Integer> list = new ArrayList<Integer>();
    show(list);
}
public static void show(ArrayList<Number> list){
    //....
}
```