

---

## Java 开发工程师笔试试题

( 请不要在试题上留任何痕迹，所有答案均写在答题纸上 )

### 一 . 编程题 ( 共 26 分 )

1. 任意写出一种排序算法。 ( 6 分 )

```
public void sort(int [] array){  
    // 代码区  
}
```

2. 求  $1+2+3+...n$  ( 不能使用乘法、 `for` 、 `while` 、 `if` 、 `else` 、 `switch` 、 `case` 等关键字以及条件判断语句 ) ( 8 分 )

```
public int sum(int n){  
    // 代码区  
    return 0;  
}
```

3. 完成下面方法，输入一个整数，输出如下指定样式图案。 ( 12 分 )

输入：3，

输出：

1\*2\*3

7\*8\*9

4\*5\*6

输入：4

输出：

1\*2\*3\*4

9\*10\*11\*12

13\*14\*15\*16

5\*6\*7\*8

```
public void drawNumPic(int n){  
    // 代码区  
}
```

### 二 . 选择题 ( 定项选择每题 3 分，不定项选择每题 4 分，共 63 分 )

1. 在基本 JAVA 类型中，如果不明确指定，整数型的默认是 \_\_ 类型，带小数的默认是 \_\_ 类型？ ( B )

A. int float  
B. int double  
C. long float  
D. long double

2. 只有实现了 \_\_ 接口的类，其对象才能序列化 ( A )

A. Serializable

- 
- B. Cloneable
  - C. Comparable
  - D. Writeable

3. 代码 `System.out.println(10 % 3 * 2)` ; 将打印出 ? ( B )  
A. 1    B.2    C.4    D.6

4. 以下程序运行的结果为 ( A )

```
public class Example extends Thread{
    @Override
    public void run(){
        try {
            Thread.sleep(1000);
        } catch (InterruptedException e){
            e.printStackTrace();
        }
        System.out.print("run" );
    }
    public static void main(String[] args){
        Example example= new Example();
        example.run();
        System.out.print("main" );
    }
}
```

- A. run main
- B. main run
- C. main
- D. run
- E. 不能确定

5. 下面有关 java 实例变量 , 局部变量 , 类变量和 final 变量的说法 , 错误的是 ? ( B )

- A. 实例变量指的是类中定义的变量 , 即类成员变量 , 如果没有初始化 , 会有默认值
- B. 局部变量指的是在方法中定义的变量 , 如果没有初始化 , 会有默认值
- C. 类变量指的是用 static 修饰的属性
- D. final 变量指的是用 final 修饰的变量

6. 在如下所示的类 Test 中 , 共有 ( D ) 个构造方法。

```
public class Test{
    private int x;
    public Test(){
        x = 35;
    }
    public void Test( double f){
```

---

```
        this .x = ( int )f;
    }
    public Test( double f){
        this .x = ( int )f;
    }
    public Test(String s) {}
}
```

A. 0      B.1      C.2      D.3

7. 下列选项中关于 Java 中 super 关键字的说法正确的是 ( A )

- A. super 关键字是在子类对象内部指代其父类对象的引用
- B. super 关键字不仅可以指代子类的直接父类，还可以指代父类的父类
- C. 子类通过 super 关键字只能调用父类的方法，而不能调用父类的属性
- D. 子类通过 super 关键字只能调用父类的属性，而不能调用父类的方法

8. Java 数据库连接库 JDBC用到哪种设计模式 ? ( B )

- A. 单例模式
- B. 桥接模式
- C. 抽象工厂
- D. 装饰模式
- E. 责任链模式

9. ( 不定项选择 ) 以下关于 JAVA语言异常处理描述正确的有? ( CD)

- A. throw 关键字可以在方法上声明该方法要抛出的异常。
- B. throws 用于抛出异常对象。
- C. try 是用于检测被包住的语句块是否出现异常，如果有异常，则抛出异常，并执行 catch 语句。
- D. finally 语句块是不管有没有出现异常都要执行的内容。
- E. 在 try 块中不可以抛出异常

10. ( 不定项选择 ) Java 多线程有几种实现方法? ( AB )

- A. 继承 Thread 类
- B. 实现 Runnable 接口
- C. 实现 Thread 接口
- D. 以上都不正确

11. ( 不定项选择 ) 下面有关 java threadlocal 说法正确的有? ( ABCD)

- A. ThreadLocal 存放的值是线程封闭，线程间互斥的，主要用于线程内共享一些数据，避免通过参数来传递
- B. 线程的角度看，每个线程都保持一个对其线程局部变量副本的隐式引用，只要线程是活动的并且 ThreadLocal 实例是可访问的；在线程消失之后，其线程局部实例的所有副本都会被垃圾回收
- C. 在 Thread 类中有一个 Map，用于存储每一个线程的变量的副本
- D. 对于多线程资源共享的问题，同步机制采用了 ‘以时间换空间’ 的方式，而 ThreadLocal 采用了 ‘以空间换时间’ 的方式

---

12. 放在 HTML里的哪一部分 JavaScripts 会在页面加载的时候被执行？( D)

- A. 文件头部位置
- B. 文件尾
- C. <head> 标签部分
- D. <body> 标签部分

13. 要在 10 秒后调用 checkState , 下列哪个是正确的 ( B)

- A. window.setTimeout(checkState, 10);
- B. window.setTimeout(checkState, 10000);
- C. window.setTimeout(checkState(), 10);
- D. window.setTimeout(checkState(), 10000);

14. 下面关于 CSS布局的描述, 不正确的是? ( D)

- A. 块级元素实际占用的宽度与它的 width 属性有关;
- B. 块级元素实际占用的宽度与它的 border 属性有关;
- C. 块级元素实际占用的宽度与它的 padding 属性有关;
- D. 块级元素实际占用的宽度与它的 background 属性有关。

15. 下面有关 javascript 常见事件的触发情况, 描述错误的是? ( C)

- A. onmousedown: 某个鼠标按键被按下
- B. onkeypress : 某个键盘的键被按下或按住
- C. onblur : 元素获得焦点
- D. onchange : 用户改变域的内容

16. head 标签中必不可少的是 ( C)

- A. <meta>
- B. <link>
- C. <title>
- D. <script>

17. 数据库有一个试卷表, 希望找出试卷平均得分小于 90 的所有试卷 ( B )

- A. SELECT \* FROM paper WHERE sum(score) < 90;
- B. SELECT \* FROM paper WHERE avg(score) < 90;
- C. SELECT \* FROM paper WHERE max(score) < 90;
- D. SELECT \* FROM paper WHERE mid(score) < 90;

18. 关于索引 (index) 的说法哪些是错误 ? ( A)

- A. 创建索引能提高数据插入的性能
- B. 索引应该根据具体的检索需求来创建, 在选择性好的列上创建索引
- C. 索引并非越多越好
- D. 建立索引可使检索操作更迅速

19. 有关数据冗余说法错误的是 ( C)

- A. 数据库中, 数据存在副本的现象, 就是数据冗余

- B. 通过分类存储，可以有效减少数据冗余，但是会增加数据查找的复杂性
- C. 在数据库设计阶段，一定要尽最大可能避免数据冗余，最好做到无数据冗余
- D. 数据冗余通常是由于数据库设计引起的

20. 在视图上使用 INSERT语句，下列 () 情况可以进行插入操作 ( C )

- A. 视图中包含了使用统计函数的结果
- B. 视图由多个表连接而成，对多个表的列进行插入操作
- C. 视图全部包含了基本表中属性为 NOT NULL的列
- D. 视图中使用了 DISTINCT

### 三 . 程序读写题 ( 共 11 分 )

1. 阅读下列程序，其输出为： 0eeE ( 5 分 )

```
public class ExceptionTest {  
  
    private static void test( int []arr) {  
        for ( int i = 0; i < arr.length ; i++) {  
            try {  
                if (arr[i] % 2 != 0) {  
                    throw new NullPointerException();  
                } else {  
                    System.out.print(i);  
                }  
            } finally {  
                System.out.print( "e" );  
            }  
        }  
    }  
  
    public static void main(String[]args) {  
        try {  
            test ( new int [] {0, 1, 2, 3, 4, 5});  
        } catch (Exception e) {  
            System.out.print( "E" );  
        }  
    }  
}
```

2. 阅读下列程序，其输出为： \_\_\_\_\_ ( 6 分 )

```
public class Aclass {  
    public Aclass(){  
        System.out.println( "A class" );  
    }  
  
    public void doPrint(){  
        System.out.println( "printer in A class" );  
    }  
}
```

```

}
public class Bclass extends Aclass{
    public Bclass(){
        System. out.println(    "B class"    );
    }
    public void doPrint(){
        System. out.println(    "printer in B class"    );
    }
}
public class Cclass extends Bclass{
    public Cclass(){
        System. out.println(    "C class"    );
    }
    public static void main(String[] args) {
        Aclass c =    new Cclass();
        c.doPrint();
    }
}

```

输出为：

A class

B class

C class

printer in B class

#### 四．简答题（共 50 分）

##### 1. String 与 StringBuffer 有什么区别？（5 分）

简单地说，就是一个变量和常量的关系。StringBuffer 对象的内容可以修改；而 String 对象一旦产生后就不可以被修改，重新赋值其实是两个对象。

StringBuffer 的内部实现方式和 String 不同，StringBuffer 在进行字符串处理时，不生成新的对象，在内存使用上要优于 String 类。所以在实际使用时，如果经常需要对一个字符串进行修改，例如插入、删除等操作，使用 StringBuffer 要更加适合一些。

##### 2. 接口和抽象类的区别是什么？（5 分）

Java 提供和支持创建抽象类和接口。它们的实现有共同点，不同点在于：

接口中所有的方法隐含的都是抽象的。而抽象类则可以同时包含抽象和非抽象的方法。

类可以实现多个接口，但是只能继承一个抽象类

类可以不实现抽象类和接口声明的所有方法，当然，在这种情况下，类也必须得声明成是抽象的。

抽象类可以在不提供接口方法实现的情况下实现接口。

Java 接口中声明的变量默认都是 final 的。抽象类可以包含非 final 的变量。

Java 接口中的成员函数默认是 public 的。抽象类的成员函数可以是 private ， protected 或者是 public 。

接口是绝对抽象的，不可以被实例化。抽象类也不可以被实例化，但是，如果它包含 main 方法的话是可以被调用的。

##### 3. 什么是 cookie ？ session 和 cookie 有什么区别？（8 分）

cookie 是 Web 服务器发送给浏览器的一块信息。浏览器会在本地文件中给每一个 Web 服务器存储 cookie。以后浏览器在给特定的 Web 服务器发请求的时候，同时会发送所有为该服务器存储的 cookie。下面列出了 session 和 cookie 的区别：

无论客户端浏览器做怎么样的设置，session 都应该能正常工作。客户端可以选择禁用 cookie，但是，session 仍然是能够工作的，因为客户端无法禁用服务端的 session。

在存储的数据量方面 session 和 cookies 也是不一样的。session 能够存储任意的 Java 对象，cookie 只能存储 String 类型的对象。

#### 4. sendRedirect() 和 forward() 方法有什么区别？（6 分）

sendRedirect() 方法会创建一个新的请求，而 forward() 方法只是把请求转发到一个新的目标上。重定向 (redirect) 以后，之前请求作用域范围以内的对象就失效了，因为会产生一个新的请求，而转发 (forwarding) 以后，之前请求作用域范围以内的对象还是能访问的。一般认为 sendRedirect() 比 forward() 要慢。

#### 5. 根据你的理解，解释一下 MVC（8 分）

（1）模型 Model：应用对象。

模型是应用程序的主体部分。模型代表了业务数据和业务逻辑；当数据发生改变时，它要负责通知视图部分；一个模型能为多个视图提供数据。由于同一个模型可以被多个视图重用，所以提高了应用的可重用性。

（2）视图 View：数据的展现。

视图是用户看到并与之交互的界面。视图向用户显示相关的数据，并能接收用户的输入数据，但是它并不进行任何实际的业务处理。视图可以向模型查询业务状态，但不能改变模型。视图还能接受模型发出的数据更新事件，从而对用户界面进行同步更新。

（3）控制器 Controller：逻辑处理、控制实体数据在视图上展示、调用模型处理业务请求。

当 Web 用户单击 Web 页面中的提交按钮来发送 HTML 表单时，控制器接收请求并调用相应的模型组件去处理请求，然后调用相应的视图来显示模型返回的数据。

#### 6. 列举并简单说明 Hibernate 实体对象生命周期的三种状态。（8 分）

（1）Transient（瞬态）：一个实体通过 new 操作符创建后，没有和 Hibernate 的 Session 建立关系，也没有手动赋值过该实体的持久化标识（持久化标识可以认为映射表的主键）。此时该实体中的任何属性的更新都不会反映到数据库表中。

（2）persistent（持久态）：当一个实体和 Hibernate 的 Session 创建了关系，并获取了持久化标识，而且在 Hibernate 的 Session 生命周期内存在。此时针对该实体任何属性的更改都会直接影响到数据库表中一条记录对应字段的更新，也即与对应数据库表保持同步。

（3）Detached（游历态）：当一个实体和 Hibernate 的 Session 创建了关系，并获取了持久化标识，而此时 Hibernate 的 Session 的生命周期结束，实体的持久化标识没有被改动过。针对该实体的任何属性的修改都不会及时反映到数据库表中。

#### 7. 介绍一种你比较熟悉的设计模式。（10 分）