第 25 期期中测试题

注：请在文本文件中写出试题的题目及答案

一、问答题

1. 说说你对 java 这门语言的理解

答：自由发挥。和 C 语言不同，java 是一门面向对象语言，去掉了很多 C 语言的复杂特性使得编程更加简洁高效。除了支持 8 种基本数据类型之外使用类来定义对象的各种行为，同时支持继承和多态减少程序设计的复杂性。Java 还支持跨平台，一次编写到处运行，这得益于在任意操作系统上都对应的 java 虚拟机（jvm），通过 jvm 执行编译过的字节码文件（.class 文件）即可实现跨平台，而不需要程序员为每个平台单独编写代码。

2. Jdk、jre、jvm 有什么区别和联系

答：JDK 是 java 开发工具包，JRE 是 java 运行时环境，JVM 是 java 虚拟机。

Java 开发工具包是 Java 环境的核心组件，并提供编译、调试和运行一个 Java 程序所需的所有工具，可执行文件和二进制文件。JDK 是一个平台特定的软件，有针对 Windows，Mac 和 Unix 系统的不同的安装包。可以说 JDK 是 JRE 的超集，它包含了 JRE 的 Java 编译器，调试器和核心类。

Java 虚拟机(JVM)

JVM 是 Java 编程语言的核心。当我们运行一个程序时，JVM 负责将字节码转换为特定机器代码。JVM 也是平台特定的，并提供核心的 Java 方法，例如内存管理、垃圾回收和安全机制等。JVM 之所以被称为虚拟的是因为它提供了一个不依赖于底层操作系统和机器硬件的接口，这种独立于硬件和操作系统的特性正是

Java 程序可以一次编写多处执行的原因。

Java 运行时环境(JRE)

JRE 是 JVM 的实施实现，它提供了运行 Java 程序的平台。JRE 包含了 JVM、Java 二进制文件和其它成功执行程序的类文件。JRE 不包含任何像 Java 编译器、调试器之类的开发工具。如果你只是想要执行 Java 程序，你只需安装 JRE 即可.

综上所述，JDK 是用于开发的而 JRE 是用于运行 Java 程序的。JDK 和 JRE 都包含了 JVM，从而使得我们可以运行 Java 程序。JVM 是 Java 编程语言的核心并且具有平台独立性。

3. 说出目前所知道的创建类的对象的方式

答：1) 使用 new 关键字，调用类的构造方法。2) 使用反射机制，调用 newInstance 方法

4. 阐述实例变量和静态变量的区别，并说明静态方法、静态变量、静态代码快、属性、普通方法之间的相互调用方式

答：静态变量是被 static 修饰符修饰的变量实例变量对于类的实例（对象）来说，都拥有自己的的一份，每个对象可以有不同的实例变量值。但是静态变量对于类的所有对象只有一份，所有对象的静态变量的值是相同的。静态方法可以调用静态变量和静态方法，不可以调用属性和普通方法；静态代码快可以调用静态变量和静态方法，不能调用属性和普通方法；普通方法可以调用静态方法、静态变量、属性和普通方法。 静态代码快只有在第一次使用类的时候执行，并且只执行一次。

5. 给出以下表达式的值:

a) (0 + 15) / 2

b) 2.0e-6 * 100000000.1

c) true && false || true && true

答：7 //类型是整形，所以输出 7

200.0000002 //浮点型

true //false||true，输出

6. Overload 和 Override 的区别，构造器是否可以 Override?

答：Overload 重载，Override 重写。重写出现在继承关系中，子类可以根据自己的需要重写父类中的方法，当然父类中 private 修饰和 final 修饰的方法不能被重写，重写后，子类对象调用该方法时就会执行子类中重写的方法，而不会再调用父类中对应的方法。重载指在一个类中，方法名相同，但是参数列表不同的多个方法，参数列表不同是指数量不同、类型不同和顺序不同。

构造器不能被继承，因此不可以重写。

7. 接口和抽象类的区别?

答：抽象类是使用 abstract 关键字修饰的类，这种类不能被实例化，并且可以有普通方法和抽象方法，抽象方法是指没有方法体的方法，需要在子类中重写得到实现。抽象类存在的意义是作为父类存在，主要用于被继承，所以类不能用 final 修饰，当一个类继承自一个抽象类，必须要实现抽象类中定义的所有抽象方法，除非该类也是一个抽象类。

接口中不能有普通方法和实例变量，接口中定义的都是抽象方法和常量，所以它比抽象类更“抽象” 。接口中定义的方法，主要用于被类实现，一个类可以实现多个接口，实现接口后，必须实现接口中定义的所有方法，除非是抽象类。接口之间可以继承，并可以多继承

8. 有两个对象 x,y 并 x.equals(y)结果为 true，则 x 和 y 的 hashcode 值是什么关系，

为什么?

答：必须相等。两个对象的 hashcode 方法返回值相同，但 equals 方法的比较不一定为 true；两个对象的 equals 方法返回值为 true，这两个对象的 hashcode 值一定相同

9. 最有效的计算 2*8 的方法

答：2<<3

10. String 和 StringBuilder、StringBuffer 的区别

答：String 类在 java 中表示字符串，是引用数据类型、 常量，在创建之后值不能更改。 因此，字符串多次累加会造成性能低下，StringBuffed、 StringBuilder 是为了解决该问题而出现的。 StringBuffer 是线程安全的类，而 StringBuilder 是非线程安全的

11. String s=new String(“xyz”);创建了几个字符串对象?

答：一个是静态存储区的"xyz",一个是用 new 创建在堆上的对象。

12. 谈谈 java 序列化

答: 将 java 对象写入流的过程称为序列化，将对象从流中读取出来称为反序列化。参加序列化的类必须要实现 java.io.Serializeble 接口，并生成序列化 ID。静态属性和使用 transient 关键字修饰的属性不参与序列化。如果父类是可序列化的，那么子类必然是可序列化的。类序列化时，属性依赖的对象必须是可序列化的对象。

13. 集合框架 List、Map、Set 三个接口有什么区别

答: List、 Set 是 Collection 接口的子接口。 List 集合中所有的元素是有序并且允许重复元素；而 Set 集合中不允许存在重复元素，有的 Set 接口实现类中元素有序（TreeSet），有的时无序的（HashSet）；Map 是不同的继承体系，把键(key)

映射到值(value)的对象，键不能重复

14. HashMap 和 Hashtable 有什么区别

答：HashMap 是 Map 接口的实现类，允许 null 键和 null 值的存在，无序，非线程安全。Hashtable 是 Map 接口的实现类，和 HashMap 最大的区别就是线程安全，在单线程的情况下，性能相对 HashMap 较低

15. ArrayList 、Vector 和 LinkedList 有什么区别

答：ArrayList 是大小可变的允许 null 值的非线程安全的 List 接口的实现类和 Vector 的区别就是 Vector 是线程安全的。LinkedList 是 List 接口的链表实现，每一个元素都和它的前一个和后一个元素链接在一起，实现所有可选的列表操作，并且允许所有元素（包括 null）。除了实现 List 接口外，LinkedList 类还为在列表的开头及结尾 get、remove 和 insert 元素提供了统一的命名方法。这些操作允许将链接列表用作堆栈、队列或双端队列。LinkedList 也是非线程安全

16. HashSet 和 TreeSet 有什么区别

答:二者都是 Set 集合的实现类,元素不可重复,都是非线程安全。区别就是 HashSet 是无序的，TreeSet 是有序的

17. 线程的生命周期

答：新建—>就绪—>运行—>死亡

新建—>就绪—>运行—>（阻塞—>就绪—>运行）—>死亡

18. 如何引起线程安全问题，如何解决

答：多线程的情况下，多个线程访问同一个对象的实例变量会引起线程安全问题。第一种在方法的声明处添加 synchronized 关键字；第二种使用 synchronized 同步代码块；第三种使用 ReentrantLock 同步锁。

19. 创建线程的安全的 ArrayList 集合

答: List<String> list = Collections.synchronizedList(new ArrayList<String>());

20. 写出五种常见的 runtimeException

答： NullPointerException 、 ClassCastException 、 ArithmeticException 、 IndexOutOfBoundsException、NoSuchElementException

21. 说明三大范式

答：1） 确保每列的原子性

2） 在第一范式的基础上，确保每列都和主键相关

3） 在第二范式的基础上，确保每列都和主键直接相关，而不是间接相关

22. Having 和 where 的区别

答：where 用于分组前过滤，having 用于分组后的过滤

23. PreparedStatement 和 Statement 有什么区别

答：1）PreparedStatement 是预编译的，可以提高性能；2）PreparedStatement 可以解决 sql 漏洞注入问题

24. 什么是数据库连接池？有什么作用

答：由于创建连接和释放连接都有很大的开销（尤其是数据库服务器不在本地时，每次建立连接都需要进行 TCP 的三次握手，再加上网络延迟，造成的开销是不可忽视的），为了提升系统访问数据库的性能，可以事先创建若干连接置于连接池中，需要时直接从连接池获取，使用结束时归还连接池而不必关闭连接，从而避免频繁创建和释放连接所造成的开销，这是典型的用空间换取时间的策略（浪费了空间存储连接，但节省了创建和释放连接的时间）。池化技术在 Java 中是很常见的，在使用线程时创建线程池的道理与此相同。据库连接池常用的有： C3P0、DBCP 等。

25. 事务的四大特性是什么

答：1)原子性(Atomic)：事务中各项操作，要么全做要么全不做，任何一项操作的失败都会导致整个事务的失败；

2)一致性(Consistent)：事务结束后系统状态是一致的；

3)隔离性(Isolated)：并发执行的事务彼此无法看到对方的中间状态；

4)持久性(Durable)：事务完成后所做的改动都会被持久化，即使发生灾难性的失败。通过日志和同步备份可以在

故障发生后重建数据。

26. 常见的 HTTP 服务器和 Servlet 容器有哪些，二者的区别是什么?

答:常见的 HTTP 服务器为 apache、Nigix,常见的 Servlet 容器为 tomcat、jetty;

Http 服务器只可以处理 Http 请求， Servlet 容器是一种程序服务器,实现了 servlet 和 jsp,同时具有 Http 的功能

27. 列出所知道的 JSP 的内置对象（共 9 个）

答：request 用户端请求，此请求会包含来自 GET/POST 请求的参数

- response 网页传回用户端的回应
- pageContext 网页的属性是在这里管理
- session 与请求有关的会话期
- application servlet 正在执行的内容
- out 用来传送回应的输出
- config servlet 的构架部件
- page JSP 网页本身
- exception 针对错误网页，未捕捉的例外

常用的组件：request、response、out、session、application、exception

28. Jsp 指令包括哪些?

答：taglib include page

29. Http 请求的特点

答：响应式、断开式、无状态协议

30. 描述 jsp 和 setvlet 的区别和联系

答：JSP 在本质上就是 SERVLET,但是两者的创建方式不一样.Servlet 完全是 JAVA 程序代码构成，擅长于流程控制和事务处理，通过 Servlet 来生成动态网页很不直观.JSP 由 HTML 代码和 JSP 标签构成，可以方便地编写动态网页.因此在实际应用中采用 Servlet 来控制业务流程，而采用 JSP 来生成动态网页

31. Jsp 文件的执行过程

答：(1) 客户端向服务器发送请求;

(2) 服务器接受到客户端的请求后将 JSP 文件编译为.java 文件;

(3) 将.java 文件并编译为 class 文件;

(4) 执行该文件并响应给客户端;

(5) 第二次加载则会找到对应的.class 文件直接调用执行,如果修改了 jsp 文件则还需要编译执行

32. Servlet 的执行过程

答：(1) 客户端发出请求

(2) 容器(Tomcat)生成 request 和 response 对象

(3) 容器根据 URL 找到合适的 Servlet 并分配线程

(4) 访问 service 方法根据请求头决定调用 doGet 或 doPost

(5) Servlet 使用响应对象通过容器给客户端做出响应,service 方法执行结束,访问对线程和 request,response 对象被销毁

33. Servlet 生命周期

答：(1) 实例化:web 容器创建 Servlet 的实例;

(2) 初始化:web 容器调用 Servlet 的 init()

(3) 服务:由 service()决定调用的是 doGet()还是 doPost();

(4) 销毁:web 容器在销毁 Servlet 之前调用 destroy();

34. 请求转发和重定向跳转的区别

答：

重定向

(1)重定向跳转使用 URL 重写的方式进行值的传递,值显示在浏览器地址栏中

(2) 重定向适合传递不敏感的数据以及简单的字符串数字等基本数据类型

(3) 重定向跳转后浏览器的地址显示的是跳转目标的 URL（地址栏发生了改变）

(4) 重定向不会引起表单的重复提交

(5) 重定向本质上是服务器产生 302 响应，在消息报文中增加 location 键值对，客户端获得 location 的值并向服务器发起第二次请求。

(6) 实现方式调用 request 的 sendRedirect()方法,不可以通过 setAttribute()设置值

请求转发：

(1) 请求转发使用 HttpServletRequest 对象的 setAttribute 方法进行值传递，值不会显示在地址栏中

(2) 请求转发跳转传值方式适合传递敏感数据以及对象以及对象、数组、集合

等类型的数据

(3) 请求转发跳转后地址栏不会显示跳转的目标的 URL(地址栏不会发生改变)

(4) 请求转发跳转会引起表单的重复提交

(5) WEB-INF 下的 jsp 页面不能被客户端直接访问，只能通过服务器跳转访问，因此访问页面只能通过请求转发跳转访问，不能通过重定向（本质是客户端请求）

(6) 实现方式是调用 request 的 getRequestDispatch.forward()方法

35. 什么是 MVC 模式

答：(1) M(Model) 模型：处理业务逻辑

(2) V(View)视图层：显示查询结果、收集用户数据

(3) C(Controller)控制器层：接收 view 请求并将请求转交给对应的 Model 并向客户

端做出相应

(4) 实现了代码的分离,降低耦合度

36. 简述在服务端如何判断用户的登录状态

答：1)客户端和服务器产生请求响应时，服务端会开辟一块内存（session）用于存储客户端的数据，并产生唯一的 sessionID 来标识 session 空间并将 sessionID 返回给客户端；客户端第二次请求就会在请求头中添加 sessionID 找到对应的内存空间 2）客户端第一次登录成功后，将对象(例如 admin 对象)存储到 session 空间中，并返回登录成功 3）客户端访问其他请求需要校验是否登录的时候，可以根据客户端传过来的 sessionID 找到对应的 session 空间,查找是否存在 admin 对象，如果不存在则重新登录。

37. 简述如何开发记住账户的功能

答：1）通过 req.getParameter(“rememberme”)的值来判断用户是否选择记住账户

2）如果记住账户，则将账户名以键值对的形式存储到 Cookie 中

3）下次在访问登录页面时，先将 cookie 中的账户名的 value 获取到，并设置到 request 的响应中，并在前端显示到页面上

38. 简述解决表单重复提交的两种方式

答：1)PRG

2)token 令牌机制

39. 如何获得请求的路径

答：request.getRequestURI()

40. 简述文件上传的步骤

答：前端页面：

(1) input 框 type 设置为 file

(2) method 设置为 post

(3) 增加 enctype="multipart/form-data"属性

后端服务:

(1) 创建上传路径

(2) 创建上传临时路径

(3) 创建 DiskFileItemFactory factory = new DiskFileItemFactory()，并设置属性：

i. 设置缓冲区大小 factory.setSizeThreshold(1024 * 10);

ii. 设置临时文件夹 factory.setRepository(tempFile);

iii. 创建 ServletFileUpload 对象并设置最大文件大小：

1. ServletFileUpload fileupload = new ServletFileUpload(factory);

2. fileupload.setFileSizeMax(1024 * 1024 * 10);

(4) 获得：List<FileItem> itemList = fileupload.parseRequest(req)

(5) 迭代该集合，分为普通元素和文件元素，分开处理

(6) 文件元素：获得文件名 item.getName()，防止重名生成新 name，创建输入输出流，并将输入流拷贝到输出流 IOUtils.copy(input, output);

(7) 关闭流

41. 简述文件下载的步骤

答：前端页面：

a 标签即可：下载图片

后端服务：

(1) 获得 fileName 并创建 File 对象 new File(saveDir,fileName)；

(2) 获得 downloadName 的值，下载文件的显示名称

(3) 设置 MIME 类型 resp.setContentType("application/octet-stream");

(4) 设置文件名，添加响应头 attachment; filename="hello.jpg"

resp.addHeader("Content-Disposition", "attachment; filename=\"" + downloadName + "\"");

(5) 创建输入输出流，并将输入流拷贝到输出流 IOUtils.copy(input, output);

(6) 关闭流

42. 如何避免 ajax 的 get 缓存

答：(1) 可以加入无意义的参数,例如时间戳

(2) 可以在请求后加入一个随机数

(3) 在后端中添加响应头 no-cache

43. Xml 中如何转义

答：1) HTML 转义 2) <![CDATA[要转义的文本]]>

44. Ajax 的跨域请求

答：(1) 使用代理

(2) JSONP(只支持 GET 请求)

45. 什么是 maven 的坐标

答：(1) GroupId:定义当前 Maven 项目的实际项目

(2) ArtifactId: 定义实际项目中的一个 Maven 项目或模块

(3) Version:定义 Maven 项目当前的版本

46. Maven 查找依赖的顺序是?

答：本地仓库>私有仓库>公有仓库

47. 启动 tomcat 出现如下错误的原因是

```
Caused by: java.net.BindException: Address already in use: JVM_Bind <null>:8080
        at org.apache.tomcat.util.net.JIoEndpoint.bind(JIoEndpoint.java:407)
        at org.apache.tomcat.util.net.AbstractEndpoint.init(AbstractEndpoint.java:623)
        at org.apache.coyote.AbstractProtocol.init(AbstractProtocol.java:434)
        at org.apache.coyote.http11.AbstractHttp11JsseProtocol.init(AbstractHttp11JsseProtocol.java:119)
        at org.apache.catalina.connector.Connector.initInternal(Connector.java:981)
        ... 31 more
Caused by: java.net.BindException: Address already in use: JVM_Bind
        at java.net.DualStackPlainSocketImpl.bind0(Native Method)
        at java.net.DualStackPlainSocketImpl.socketBind(Unknown Source)
        at java.net.AbstractPlainSocketImpl.bind(Unknown Source)
        at java.net.PlainSocketImpl.bind(Unknown Source)
        at java.net.ServerSocket.bind(Unknown Source)
        at java.net.ServerSocket.<init>(Unknown Source)
        at java.net.ServerSocket.<init>(Unknown Source)
        at org.apache.tomcat.util.net.DefaultServerSocketFactory.createSocket(DefaultServerSocketFactory.
        at org.apache.tomcat.util.net.JIoEndpoint.bind(JIoEndpoint.java:394)
        ... 35 more
```

答：启动了多个端口号相同的 tomcat 容器

48. 分别给出以下代码段打印出的值:

```
int sum = 0;
for (int i = 1; i < 1000; i++)
    for (int j = 0; j < i; j++)
        sum++;
System.out.println(sum);
```

```
int sum = 0;
for (int i = 1; i < 1000; i *= 2)
    for (int j = 0; j < 1000; j++)
        sum++;
System.out.println(sum);
```

答：499500

10000

49. 给出以下表达式的值:

a) System.out.println('b');

b) System.out.println('b' + 'c');

c) System.out.println((char) ('a' + 4));

答：b

197

e

50. 给出 exR1(6) 的返回值:

```
public static String exR1(int n){
    if (n <= 0) {
        return "";
    }
    return exR1(n-3) + n + exR1(n-2) + n;
}
```

答：311361142246

二、编程题

1. **输出 10！的值(递归)。**
2. 判断一个数是质数。质数又称素数，在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数，这样的数称为质数，有无限个。
3. 编写查询的 jdbc 代码。
4. 从控制台获取一组以“,” 分隔开的数字，将这数组从大到小排序后，再以“,”分隔开。
5. 写出单例模式的两种形式