非静态内部类可以实现哪种接口

非静态内部类可以实现externalizabl comparable serializable 这些接口吗 comparable

Runtime.exec()的 api调用，使用不当容易造成os命令注入

安卓 华为 其他商业组织 第三方

java的基本类型有哪些

当方法遇到异常又不知如何处理时，该怎么办？a 捕获异常 2抛出异常 3嵌套异常 4声明异常

下列哪种情况下，需要使用sslsocket套接字 a套接字上传输的数据不敏感 2数据虽然敏感，但是已经过恰当加密 3套接字的网络路径从来不越出信任边界 4不安全的传输通道中传输敏感数据 4不安全的传输通道中传输敏感数据

java代码的列限制为多少个字符 120、

研发需要对自己的代码 代码走查

路径覆盖是最强的覆盖准则

以下哪个不属于java对象的hashcode方法的约定 a如果两个对象调用equals方法是相等，则这两个对象的hashcode方法，也必须返回相同的值 b同一次运行中，用一个对象如果equals方法中用到的信息没有改变，多次调用hashcode方法返回值必须相同 c与常亮比较 优先java.utio.objects#equals;与变量比较，优先"test".equals(object) d如果两个对象调用equals方法时不相等，则这两个对象的hashcode方法，不要求其返回值不用 但是出于性能考虑，但最好能不同 c与常亮比较 优先java.utio.objects#equals;与变量比较，优先"test".equals(object)

不需要同时覆写hashcode方法

测试数据属于测试用例的组成部分

敏感数据先数字签名再加密

同一信任安全通道内传输 不需要数字签名

网络层 -传输层 -会话层

java每行限长多少个字符

程序所包含的路径数与程序的复杂程度有关

使用不可信数据记录日志会导致 日志伪造 和 敏感信息泄露

父类中处理敏感关键信息的方法，以下哪种方式最合理 a使用public修饰 b使用final修饰 c使用protected修饰

下列说法错误的是 a 攻击者可以通过获取一个可访问对象的隐式锁并无期限持有该锁来触发条件竞争与死锁，进而引起拒绝服务（dos） b boolean类型不适合作为锁使用，因为boolean只有两个值：true和false c如果同步代码快使用静态public的锁对象，攻击者就不可能获取该锁对象 d声明为同步的方法以及在this引用上的同步快都使用对象自身的锁（隐式锁） c如果同步代码快使用静态public的锁对象，攻击者就不可能获取该锁对象

为了确保输入的数据是安全的，可以剔除或者转换某些字符（例如：删除引号 转换为html）是提前。这是什么校验方法 a白名单校验 b白名单方法净化 c黑名单校验 d黑名单方法净化

下面关于异常的说法错误的是 a不要通过一个空的catch块忽略异常 b方法抛出过多的异常，会增加客户的异常处理的工作。所以一个方法不应抛出超过6个异常。 c对可恢复的情况使用受检异常（checked exception） ，对编程错误使用运行时异常（runtine exception） d在finally块中不要使用return break或continue使finally块非正常结束

保存口令邓敏感数据应优先使用什么，然后结束后怎么样，可降低敏感数据泄露的风险 a： string，等待垃圾收集器回收 b： string，置为0 c：char[],等待垃圾收集器回收 d：char[] ,清0

transformerfactory设置什么feature可使xslt在转换xml过程中避免被注入执行任意程序恶意代码的问题 javax.xml.feature.secure-processing

推荐的防御sql注入到最佳方法是 a使用存储过程 b限制外部输入的长度 c使用参数化查询 d对外部输入进行转义

关于nullpointexception异常下面说法错误的是 a直接捕获异常，并对该异常进行处理，保证程序正确执行不会报错即可 b判断字符串是有效时不可以根据try捕获异常的方式判断其不是有效字符串 c对于调用开源三方件，三方件中抛出nullpointexception异常时，可以捕获nullpointexception，冰对该异常进行处理。 d非开源第三方件的nullpointexception异常需要对根因进行定位处理，不是只处理异常即可。

下列哪个说法是正确的 a禁止系列化非静态的内部类 b禁止系列化静态的内部类 c禁止序列化非静态的外部类 d禁止序列化静态的外部类

对内置锁的描述错误的是 a内置锁是可重入锁，如果已经获取了一个锁对象，在还没释放时又要执行该锁对象的另一个代码快或方法，则不需要再次给这个对象加锁可以直接执行 b内置锁是非公平锁，线程在竞争synchronized锁时并不遵循先到等待队列就先获得锁 c内置锁是显示锁，需要显示进行lock以及unlock操作 d内置锁是一个互斥锁，不仅读写互斥并且读读也互斥，最多只有一个线程能够获得该锁

以下关于运算和表达式说法错误的是

a在条件表达式?:的第2和第3个操作数使用同样的类型 b表达式的比较，应当遵循右侧倾向于变化，左侧倾向于不变的原则 c永括号明确表达式的操作顺序，避免过分依赖默认优先级 d不要写超过3个的复杂条件表达式

在第一行我们写一个0。接下来的每一行，将前一行中的0替换为01，1替换为10。给定行数N和序数K,返回第N行中第K个字符。（K从1开始）

例子：

输入N=1，K=1 输出：0；

输入N=2，K=1 输出：0；

输入N=2，K=2 输出：1

解释：

第一行：0

第二行：01

第三行：0110

第四行：01101001

注意：

N的范围[1，30]

K到范围[1，2^（N-1）]

根据这些要求，补全下面的java代码

import java.util.\*;

class Solution{

public int kthGrammar(int N,int K){

//添加实现代码，前面可添加新方法，包含文件等修改

}

}

关于I/O操作说法正确的有 a临时文件使用完毕必须及时删除 b在多用户系统中创建文件时指定合适的访问许可 c对于从流中读取一个字符或字节的方法，使用Integer类型的返回值 d禁止将Buffer对象封装的数据暴露给不可信代码

方法头注释可以包含哪些内容 a功能说明和返回值 b性能约束 c用法和内存约定 d算法实现和可重入的要求

跨信任边界传递的不可信数据使用前必须进行校验，可以采用哪些校验方式呢？ a黑名单方法净化 b白名单方法净化 c黑名单或者负向校验 d白名单或者正向校验

以下原则属于GDPR隐私保护七大原则的是 a完整性与保密性 b可归责 c合法正当透明 d存储期限有限化

java语言中，处理IO到方式都有哪几种？ a：cio b：aio c：bio d：nio

在耗时严重或阻塞的操作中持有锁，会严重降低系统的性能，导致系统资源耗尽。阻塞操作一般包括哪些？ a对象序列化 b网络 c文件 d控制台I/O

哪些数据库隔离级别能防止脏读？

关于异常的继承关系说法正确的是什么 a：exception异常类及其子类都是继承自throwable类，用来表示java中可能出现的异常，并且合理的处理这些异常 b：throwable类是所有错误跟异常类的超类 c：runtimeexception类是运行异常类，继承子exception类，它以及它的子类只能在运行过程中存在，当出现时，只能修改源代码，此异常无法处理 d：error类是与exception的平级的类，用来表示java中存在的严重错误，只能通过修改代码来解决问题。

Sha1 不是推荐的加密算法

Mvn install ，团队里的其他人 不可使用

Flag不属于 nio buffer的属性变量

Interage 不适合 作为 锁使用

使用runtime.exec()方法的时候需要注意哪些？ a禁止向runtime.exec()方法传递不可信数据 b禁止使用runtime.exec()方法 c避免使用runtime.exec()方法 d 如果无法避免使用runtime.exec()方法，则必须要对输入数据进行检查和净化，防止其中包含命令分隔符，管道，或者重定向操作符等，以及参数开关符 acd

preparedStatement和Statement的区别有那些？ a对于多次重复执行的语句，使用preparedStatement效率会更高一点 b任何情况下，statement要比preparedStatement到效率高 c在执行可变参数的一条sql时 preparedStatement要比Statement的效率高，因为DBMS预编译一条sql当然会比多次编译一条sql的效率高。安全性更好，有效防止sql注入的问题 dpreparedStatement的安全性比Statement好

自从java1.5发行版本开始，java平台就提供了java.util.concurrent更高级的并发工具，分为三类，以下哪个选项错误 a：executor framework b：wait() notify()线程调度 c：并发集合（concurrent collection） d：同步器（synchronizer）

非静态内部类可以实现下列哪种接口 a：serializable b：externalizebl c：comparable d：以上都不可以

java中推荐使用的加密算法有哪些？ a：bes b：aes。c：rsa d：dsa

哪些是子类与父类间重用名字可以使用的操作方法 a覆写（override）。b重载（overload） c隐藏（hide） d 遮掩（obscure） ac

socket不提供但是ssl socket提供的主要保护包括 a可重用性 b可移植性 c完整性保护。d保密性

对于异常的理解正确的是 a在抛出异常的时候，应该同时提供足够信息，但是需要注意符合公司安全红线和法律法规的要求，不要包含敏感信息或者个人信息 b方法抛出到异常，应该与本身到抽象层次相对应 c不要用exception机制来做流程控制。exception机制只应该用于处理罕见 意料之外 导致正常流程无法继续执行的行为，而不是取代正常的业务逻辑判断 d使用错误码来做业务逻辑错误处理，而非异常。

哪些类型的命名需要采用大驼峰命名 a接口 b包含 c枚举 d常量

关于泛型的说法正确的有哪些？ a数组与泛型集合相比，更灵活，应优先使用数组，而不是泛型集合 b静态工具方法很适合于泛型化。c类的设计可优先考虑泛型 d在集合中使用泛型，可以告诉编译器每个集合中接受哪些对象类型

下面的说法错误的有 a禁止在日志中保存口令 密钥和其他敏感数据 b安全场景下不需要使用强随机数 c禁止使用私有或者弱加密算法 d必须使用socket代替sslsocket来进行安全数据交互

出于性能的考虑，初始化哪些集合时，最好指定其初始化容量 a：hashmap b：hashset c：stringbuilder d：arraylist

哪个是xml格式的反序列化常见的第三方组件 a：xmldecoder b：xmlstream c：xdecoder d：xmltrans

以下关于运算和表达式说法错误的是 a不要写超过3个的复杂条件表达式 b用扣号明确表达式的操作顺序，避免过分依赖默认优先级。c在条件表达式？：的第2和第3个操作数使用同样的类型 d表达式的比较，应当遵循右侧倾向于左侧 左侧倾向于不变的原则

每行一个语句 可以提高可读性

是否能看到被测源程序 是区分黑盒测试和白盒测试的关键

类加载器的顺序 不是 ac ex bc

在 自底向上测试中，编写 测试驱动模块 来测验正在测试的模块

方法的参数个数 不应超过 5 个

命令行执行 system.exit()方法

Jstack 可用于分析 死锁 问题

Jinfo 不可以查看 java程序的内存使用情况

Transient 阻止包含敏感信息的数据成员被序列化

抽象和接口 反映的设计理念不同 错的

盐值 至少 不应该包含 16字节

字符串归一化：java.text.normalizer.noimalizer

无需索引时候采用 for-each 代替 for

能使用 不常用的双重锁定检查

如果程序依赖于线程调度器不可移植，同样程序依赖于java 的线程优先级 不可移植

白盒测试多用于 单元测试 阶段

java的哪个版本引入了可变参数 a：jdk1.5 b：jdk1.6 c：jdk1.7 d：jdk1.8

使用白盒测试方法时，确定测试数据应该根据 程序的内部逻辑 和 指定的覆盖标准

关于nullpointexception异常下面说明错误的是 a在判断字符串是有效时不可以根据try捕获异常的方式判断其不是有效字符串 b对于调用第三方件，第三方件中抛出nullpointexception异常时，可以捕获nullpointexception，并对该异常进行处理 c直接捕获异常，并对该异常进行处理 ，保证程序正常执行不会报错即可 d非开源三方件的nullpointexception异常需要对根因进行定位处理，不是只处理异常即可 c直接捕获异常，并对该异常进行处理 ，保证程序正常执行不会报错即可

需要进行精确计算时 不建议 long和double

以下哪种方式能够预防xml注入漏洞攻击 a：xml。schema校验 b：dtd校验 c：白名单校验 d：格式化字符串

下面几个类的命名符合java编码规范要求的是 a：UserAccount b:Info c:Data d:upperCamelCase

下列哪些是可靠性 可用性的分析方法 a安全 可靠测试设计分析 b基于过程markov过程 c反复实验测试

关于文件路径校验说法错误的是 a:获取文件路径时，必须使用getCanonicalPath(),禁止使用getAbsolutePath() b:File.getAbsolutePath()返回文件的相对路径 c：文件路径校验前必须先进行标准化处理 d：使用file.getcanonicalPath()方法，它能在所有的平台上对所有别名 快捷方式以及符号链接进行一致性解析

当方法遇到异常又不知如何处理时，下列哪种说法是正确的 a声明异常 b嵌套异常 c抛出异常 d捕获异常

给出一个字符串s（仅含有小写英文字母和括号） 请你按照从括号内到外的顺序，逐层反转每对匹配括号中的字符串，并返回最终到结果。注意，您的结果中不应包含任何括号

例子：

输入s="(abcd) "输出："dcba"

输入s="u(love)i"输出："iloveu"

注意：

0<=s.length <=2000

s中只有小写英文字母和括号

我们确保所有括号都是成对出现的

根据这些要求，补全下面的java代码

import java.util.\*;

class Solution{

public int reverseParentheses(String s){

//添加实现代码，前面可添加新方法，包含文件等修改

}

}

关于try-catch-finally-return的执行说法正确的有 a：finally是在return后面的表达式运行执行的，所以函数返回值在finally执行前确定的，无论finally中的代码怎么样，返回的值都不会改变，仍然是之前return语句中保存的值 b：当try和catch中有return语句时，finally块仍然会执行 c：finally中最好不要包含return，否则程序会提前退出，返回值不是try或catch中保存的返回值 d：如果有异常产生，finally块中代码不会执行