

【软考达人】

软考资料免费获取

- 1、最新软考题库
- 2、软考备考资料
- 3、考前压轴题



微信扫一扫，立马获取



6W+免费题库



免费备考资料

PC版题库：ruankaodaren.com

2023年下半年的软考终于结束了，从各个小伙伴的反馈来看，你说很难么，对于一些有实践经验的同学来说，还可以，感觉不难，但是落到纸面上，就差强人意了，平时这方面要多练习，**所想所思要落到纸面上，或者表达清晰让别人听懂，不仅是工作中一个基本素质，也是个非常重要的能力**，不仅仅在学习工作中，生活中也一样，让别人清晰知道你的需求，但是对于小白，没有相关从业经验的同学来说，一直是有难度的，下午案例必选题换成大数据了？够惊喜的，但也不意外，咱们考试之前也分析过，必选题有可能会变，还真的变了，变成大数据了，这就引来了很多小伙伴的吐槽，并不是所有的公司都需要大数据的，尤其对于中小型公司，数据体量达不到，强行上大数据是不是有点为了技术而技术的感觉，就像马斯克喊下云，别人都在喊上云，马斯克发现上了云之后人员徒增，成本也涨了，那我上云的意义是什么？

整体的趋势，随着改版，越来越与时俱进，你说它真正的偏实践么？其实也都是教材上的理论知识，实际工作中可能不一定是采用它的解决方案，毕竟实际工作中考虑的东西会更实际，所以说应付考试的话，还是要把书上的理论知识通读一遍，了解清晰。

1.上午案例知识

上午案例知识有小伙伴考记忆整理了一份（佩服记忆力）：

1-2. McCabe 度量法（错误的是 $V(g)=m-n+1$ 、最大范围<10）3. 多线程任务通信方式（共享内存+消息传递）4. 星型拓扑结构最多2跳 5. 数据库

(A->B, B->C), 属于 2NF6. 数据库去除多值依赖属于 4NF7. 不仅关注输入输出, 也关注逻辑测试: 灰盒测试 8. 数据库语句 (having+group by) 9. web 新型测试: A/B 测试 10. SSL 不具备特性 (不可抵赖性) 11-12. sysml 新增 (需求图), 其中 (需求图) 描绘了需求 13. 面向对象需求不断新增开发模型 (喷泉模型) 14. 开发和测试同时进行 (W 模型) 15. 阶段结束后输出物经过评审变为 (软件配置项) 16. 敏捷开发说法不正确的是 (以用例为中心) 17. 进度管理, 数学题 (选择 6 天) 18. 数学题, 给定序列推导, 说法不正确的是(参数取值 0.1-0.3)

19. 单 CPU 任务运行时 (任务交替运行)

20. 通道是(CPU)连接外设的

21. 软件架构复用包括机会复用和(系统复用)

22. 数据库展示给用户的是 (外模式)

23. 静态架构评估 (static)

24-25. 可靠性计算, (MTTF 与 MTBF 时间几乎相等), 包括(容错性和健壮性)

26. 不对称加密算法(RSA)

27-28. 质量属性 (性能+可移植性)

29.内核功能包括 (进程管理+文件系统管理+设备驱动管理+内存管理)

30.架构是(一组词汇表+约束)

31.需求分析常用(petri 网)

32.PDCA 指的是 (计划、执行、检查和处理)

33.(进程) 是操作系统分配资源最小单位

34.质量效用树结构 (树根-质量属性-属性分类-质量属性场景)

35-36. M2M 包含部分有 (智能化机器+M2M 硬件+通信网络+中间件
+应用其中 (应用)对获得的数据进行加工分析, 提供决策依据)

37.DSSA 说法正确的是 (因为 DSSA 领域不同, 所以使用互有差异)

38.DO178 包括 (指导、目标、活动、证据)

39.CDN 和反向代理都是基于 (缓存)

40.专利许可不包括 (特殊许可)

41.关于局域网说法错误的是 (非封闭性网)

42.(两层胖客户端+瘦服务器)

43-44.ABSD 最顶层被分解为 (概念子系统) , ABSD 需求来源(开发人员商业目标)

45.哪种类型不适合自动化测试 (需求频繁变化的)

46.嵌入式系统设计中要考虑的因素 (可升级、可配置、易于操作、接口规范、重量、功耗、成本、开发周期)

47.C 语言机器学习算法题 (blast)

48.NPU 说法正确的是 (计算速度快, 性能好)

49-50 质量属性场景刺激发生的条件 (环境), 针对性能影响度量 (吞吐量)

51.SAAM 主要输入的问题是问题描述、(需求说明)和架构描述文档

52.ABSD 方法进行软件开发时, 需要经历架构需求、架构设计、架构文档化、架构复审、架构实现和架构 (演化)

53-54.领域分析阶段获得 (领域模型)领域输出阶段获得 (特定领域软件架构)

55.数据库风格, 对批处理和管道进行比较错误的是 (管道节点并行)

56.构件接口调用其他服务是(分层, 不是叠加)

57. 变更管理顺序(提出变更申请、进行风险评估、进行变更批准、开展变更实施、变更验证、存档)

58-59. 代码覆盖率是 (单元测试), 链接测试点是 (web)

60. 区块链没有 (行业链)

61. 构件说法错误的是 (构件没有外部可见状态)

62. (构件检索)需求列表, 构件建立分类索引模式;

63-64. 构件分别是(适应性构件)和(装配性构件)

65. 构建管理包括构件描述、(构件分类)、构件库组织等

66. 专利说法错误的是 (专利一定是申请人、发明人的)

67. (契约式) 接口开发

68-69. 构件 (可组装性) 所外部接口必须通过公开接口定义, 和 (可部署性) 必须能作为一个独立实体在构件平台上运行

70. 某排序题

71-75 层次式体系结构设计是将系统组成个层次结构, 每一层为上层服务, 并作为下层客户。

在(一些 some)层次系统中，除了一些精心挑选的输出函数外，内部的层接口只对相邻的层(可见 show)。连接件通过决定层间如何交互的协议来定义

拓扑约束包括对相邻层间交互的(约束 limiting)。由于每一层最多只影响两层，同时只要给相邻层提供相同的接口，可传递的系统级(抽象 abstract)允许每层用不同的方法实现，同样为软件重用提供了强大的支持。在分层次体系结构中的组件被(划分 partition)成几个层，每个层代表应用的一个功能，都有自己特定的角色和职能。

答案仅供参考，不一定正确。

2. 下午案例知识

案例的考点：

1. 大数据架构 Lambda 架构和 Kappa 架构
2. jwt 特点
3. 数据持久层，Redis 数据丢失，数据库读写分离方案
4. Hibernat 架构
5. SysML 七个关系，填需求图

大数据的必选题：

某网作为某电视台在互联网上的大型门户入口，某一年成为某奥运会中国大陆地区的特权转播商，独家全程直播了某奥运会全部的赛事，积累了庞大稳定的用户群，这些用户在使用各类服务过程中产生了大量数据，

对这些海量数据进行分析与挖掘，将会对节目的传播及商业模式变现起到重要的作用。该奥运期间需要对增量数据在当日概览和赛事回顾两个层面上进行分析。

其中，当日概览模块需要秒级刷新直播在线人数、网站的综合浏览量、页面停留时间、视频的播放次数和平均播放时间等千万级数据量的实时信息，而传统的分布式架构采用重新计算的方式分析实时数据在不扩充以往集群规模的情况下，无法在几秒内分析出重要的信息。

赛事回顾模块需要展现自定义时间段内的历史最高在线人数、逐日播放走势、直播最高在线人数和点播视频排行等海量数据的统计信息，由于该奥运期间产生的数据通常不需要被经常索引、更新，因此要求采用不可变方式存储所有的历史数据，以保证历史数据的准确性。

[问题 1] (8 分)

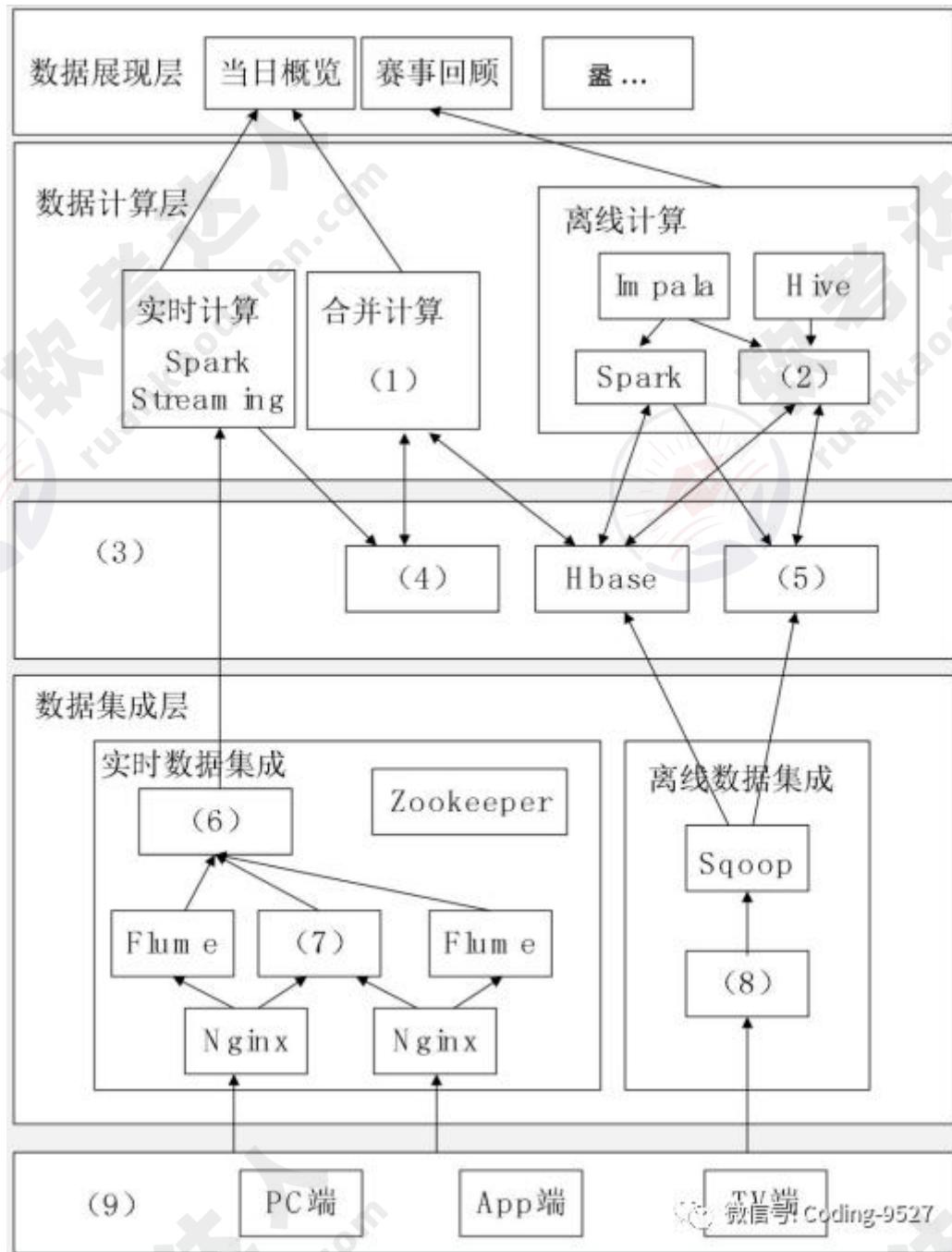
请根据 Lambda 架构和 Kappa 架构特点，填写以下表格。

对比内容	Lambda架构	Kappa架构
复杂度与开发、维护成本	需要维护 (1) 套系统（引擎），复杂度 (3) ，开发、维护成本 (3)	只需要维护 (2) 套系统（引擎），复杂度 (4) ，开发、维护成本 (4)
计算开销	需要一直运行批处理和实时计算，计算开销大	(5)
实时性	满足实时性	(6)
历史数据处理能力	批式全量处理，吞吐量 (7) ，历史数据处理能力强	流式全量处理，吞吐量相对较低， 历史数据处理能力 (8)

[问题 2] (9 分)

下图 1 给出了某网奥运的大数据架构图, 请根据下面的 (a)~(n) 的相关技术, 判断这些技术属于架构图的哪个部分, 补充完善下图 1 的 (1) - (9) 的空白处。

(a) Nginx; (b) Hbase; (c) Spark Streaming; (d) Spark; (e) MapReduce; (f) ETL; (g) MemSQL; (h) HDFS; (i) Sqoop; (j) Flume ; (k) 数据存储层; (l) kafka; (m) 业务逻辑层; (n) 数据采集层



[问题 3] (8 分, 每空 2 分)

大数据的架构包括了 Lambda 架构和 Kappa 架构, Lambda 架构分解为三层: 即(1)、(2)和(3); Kappa 架构不同于 Lambda 同时计算流计算

和批计算并合并视图，Kappa 只会通过流计算一条的数据链路计算并产生视图。

请问该系统的大数据架构是基于哪种架构搭建的大数据平台处理奥运会大规模视频网络观看数据。

【问题1】

(1) 2 (2) 1 (3) 高 (4) 低

(5) 必要时进行全量计算，计算开销相对较小

(6) 满足实时性 (7) 大 (8) 弱

【问题2】

(1) d (2) e (3) k (4) g (5) h

(6) l (7) j (8) f (9) n

【问题3】

(1) 批处理层 (2) 加速层 (3) 服务层

该系统的大数据架构是基于Lambda架构搭建的大数据平台处理奥运会大规模视频网络观看数据。

微信号: Coding-9527

JWT 全称 (JSON Web Token) , 教材没有的超纲内容。

JWT (JSON Web Token) 是一种用于在网络上安全传输声明的开放标准 (RFC 7519)。JWT 通常由三个部分组成，这三个部分分别是头部 (Header)、载荷 (Payload) 和签名 (Signature)。这三个字段通常用点号 (.) 分隔开，形成一个完整的JWT字符串。下面是这三个字段的详细介绍：

1. **头部 (Header)**：JWT头部通常由两部分组成，标识令牌类型和所使用的签名算法。例如：

```
json
Copy code
{
  "alg": "HS256",
  "typ": "JWT"
}
```

在头部中，`alg` 表示使用的签名算法，例如 HMAC SHA256 算法。`typ` 表示令牌的类型，即JWT。

2. **载荷 (Payload)**：载荷包含了一些声明 (claims)。声明是关于实体 (通常是用户) 和其他数据的声明。有三种类型的声明：注册声明、公共声明和私有声明。

注册声明 (Registered claims) 包括了一些预定义的声明，不是强制性的，但是推荐使用，包括：

- `iss` (issuer)：表示令牌的发行者。
- `sub` (subject)：表示令牌的主题，即令牌所代表的用户。
- `aud` (audience)：表示令牌的观众，即接收令牌的一方。
- `exp` (expiration time)：表示令牌的过期时间。
- `nbf` (not before)：表示令牌的生效时间。
- `iat` (issued at)：表示令牌的发行时间。
- `jti` (JWT ID)：表示JWT的唯一标识符。

微信号: [CoLg9gk527](https://mp.weixin.qq.com/s/CoLg9gk527)

公共声明 (Public claims) 是由JWT规范中未定义的声明，可以定义自己的声明内容。私有声明 (Private claims) 是用于在同意使用它们的双方之间共享信息的声明。

一个示例载荷可能是这样的：

```
json
Copy code
{
  "sub": "1234567890",
  "name": "John Doe",
  "admin": true
}
```

3. **签名 (Signature)**：要创建签名部分，你需要使用编码后的头部、编码后的载荷，以及一个秘密 (通常是一个密钥，例如服务器密钥)，然后使用指定的算法 (在头部的`alg`字段中指定) 生成签名。签名的作用是验证消息的发送者以及确保消息在传递的过程中没有被篡改。

总结起来，JWT由这三个部分组成，形成一个用点号分隔的字符串：

`Header.Payload.Signature`。这种结构使得JWT非常适合在网络上传输，轻松地解码和验证。

Hibernate 一个 java 现在都不怎么用的 ORM，属实是没有想到！不过都是教材上的内容。

Hibernate 是一个开放源代码的对象关系映射框架，它对 JDBC 进行了轻量级的对象封装，使 Java 程序员可以随心所欲地使用对象编程思维来操纵数据库。它不仅提供了从 Java 类到数据表之间的映射，还提供了数据查询和恢复机制。相对于使用 JDBC 和 SQL 来手工操作数据库， Hibernate 可以大大减少操作数据库的工作量。另外， Hibernate 可以利用代理模式来简化载入类的过程，这将大大减少利用 Hibernate QL 从数据库提取数据的代码的编写量。 Hibernate 可以和多种 Web 服务器或者应用服务器良好集成，如今已经支持几乎所有流行的数据库服务器。

Hibernate 技术本质上是一个提供数据库服务的中间件，它的架构如图13-14所示。

图13-11显示了 Hibernate 件（如 `hibernate.properties`）的工作原理，它是利用数据库以及其他一些配置 XML Mapping 等来为应用程序提供数据持久化服务的。

Hibernate 具有很大的灵活性，但同时它的体系结构比较复杂，提供了好几种不同的运行方式。在轻型体系中，应用程序提供 JDBC 连接，并且自行管理事务，这种方式使用了 Hibernate 的一个最小子集。在全面解决体系中，对于应用程序来说，所有底层的 JDBC/JTA API 都被抽象了，Hibernate 会照管所有的细节。

Hibernate 是一个功能强大，可以有效地进行数据库数据到业务对象的 O/R 映射方案。 Hibernate 推动了基于普通 Java 对象模型，用于映射底层数据结构的持久对象的开发。通过将持久层的生成自动扩展到一个更大的范围， Hibernate 使开发人员专心实现业务逻辑而不用分心于



图 13-14 Hibernate 架构图

3.下午论文

可靠性分析与评价方法

面向对象分析 (ooa)

多数据源

边云协同

四个论文，前两个是可以写的，估计也有很多同学在纠结写哪个，剩下两个确实有难度。