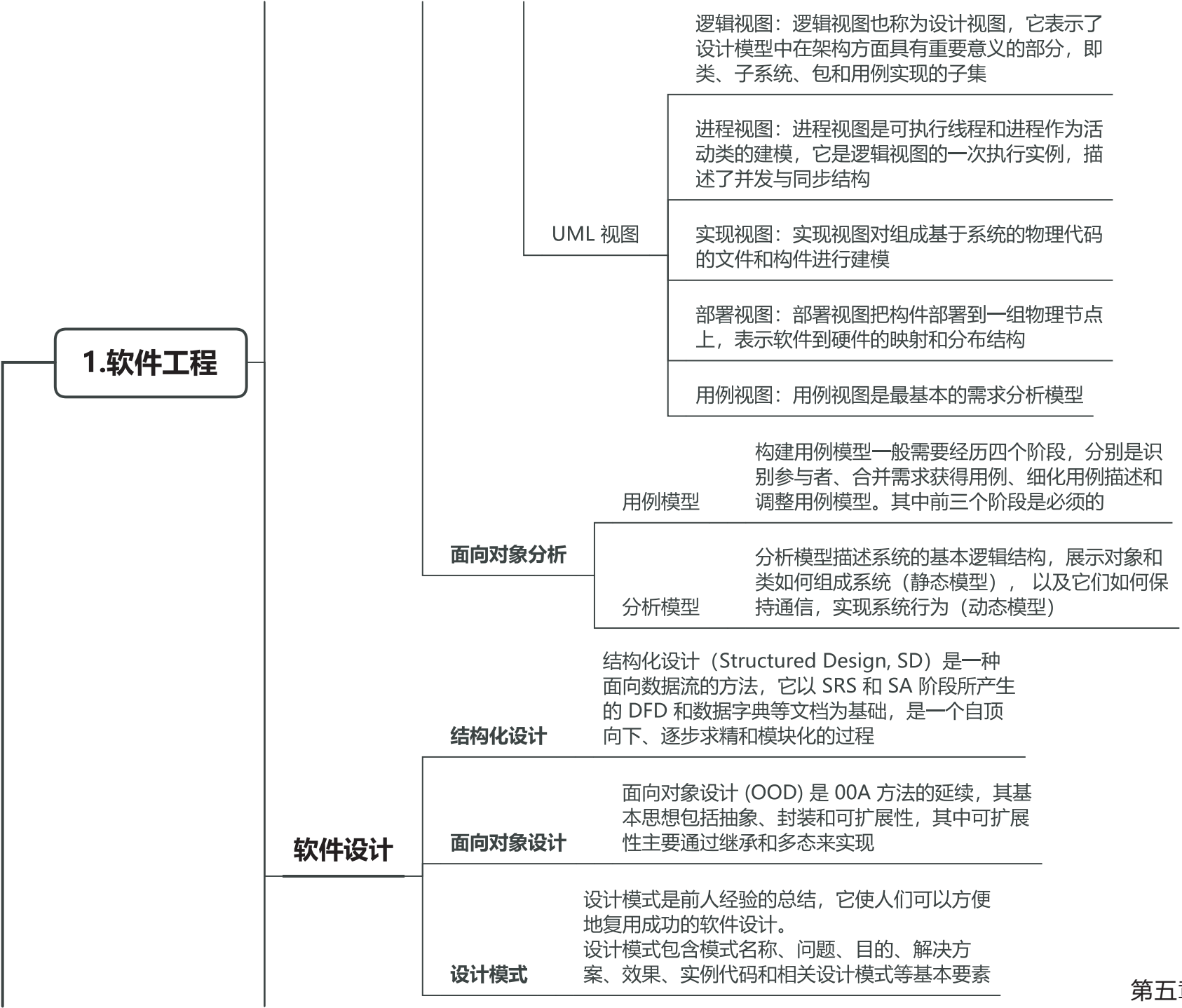


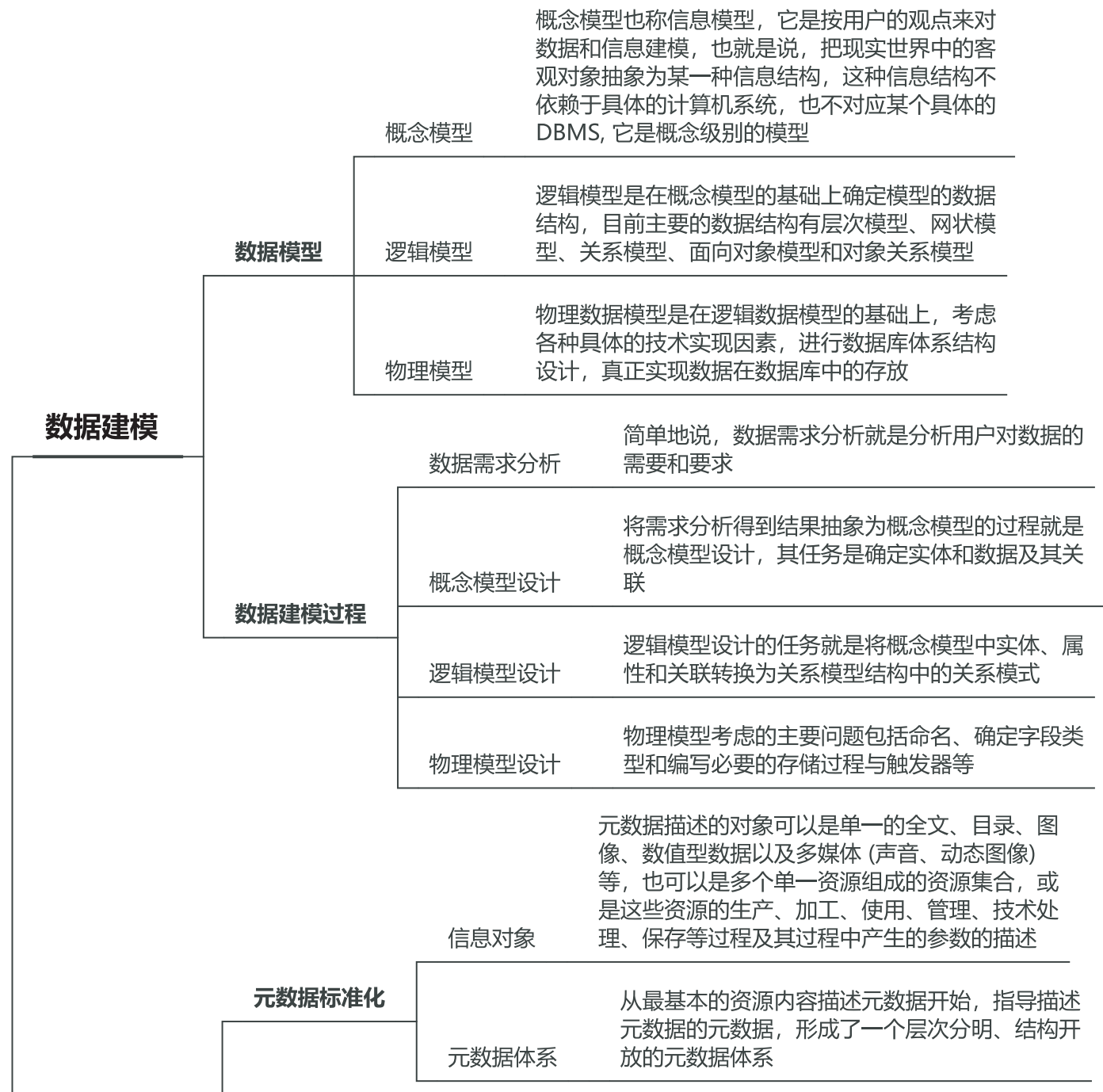
架构设计	软件架构风格		软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式(Idiomatic Paradigm)。架构风格反映了领域中众多系统所共有的结构和语义特性，并指导如何将各个构件有效地组织成一个完整的系统
	软件架构评估		软件架构评估可以只针对一个架构，也可以针对一组架构。在架构评估过程中，评估人员所关注的是系统的质量属性。从目前已有的软件架构评估技术来看，可以归纳为三类主要的评估方式，分别是基于调查问卷 (或检查表) 的方式、基于场景的方式和基于度量的方式
	需求的层次		需求是多层次的，包括业务需求、用户需求和系统需求，这三个不同层次从目标到具体，从整体到局部，从概念到细节
	需求过程	需求获取	需求获取是一个确定和理解不同的项目干系人的需求和约束的过程
		需求分析	需求分析对已经获取到的需求进行提炼、分析和审查，以确保所有的项目干系人都明白其含义并找出其中的错误、遗漏或其他不足的地方
		需求规格说明书编制	软件需求规格说明书（SRS）是需求开发活动的产物，编制该文档的目的是使项目干系人与开发团队对系统的初始规定有一个共同的理解，使之成为整个开发工作的基础
		需求验证与确认	需求验证与确认活动内容包括： <ul style="list-style-type: none"><li>•SRS正确地描述了预期的、满足项目干系人需求的系统行为和特征；</li><li>•SRS中的软件需求是从系统需求、业务规格和其他来源中正确推导而来的；</li><li>•需求是完整的和高质量的；</li><li>•需求的表示在所有地方都是一致的；</li><li>•需求为继续进行系统设计、实现和测试提供了足够的基础。</li></ul>
UML 中的事物	结构事物		结构事物在模型中属于最静态的部分，代表概念上或物理上的元素
	行为事物		行为事物是 UML 模型中的动态部分，代表时间和空间上的动作
	分组事物		分组事物是 UML 模型中组织的部分，可以把它们看成是个盒子，模型可以在其中进行分解
	注释事物		注释事物是 UML 模型的解释部分

需求分析			<p>依赖 (Dependency)：依赖是两个事物之间的语义关系，其中一个事物发生变化会影响另一个事物的语义</p>
			<p>关联 (Association)：关联描述一组对象之间连接的结构关系</p>
	UML	UML 中的关系	<p>泛化 (Generalization)：泛化是一般化和特殊化的关系，描述特殊元素的对象可替换一般元素的对象</p>
		UML2.0 中的图	<p>实现 (Realization)：实现是类之间的语义关系，其中的一个类指定了由另一个类保证执行的契约</p>
			<p>类图：描述一组类、接口、协作和它们之间的关系</p>
			<p>对象图：描述一组对象及它们之间的关系</p>
			<p>构件图：描述一个封装的类和它的接口、端口，以及由内嵌的构件和连接件构成的内部结构</p>
			<p>组合结构图：描述结构化类 (例如，构件或类) 的内部结构，包括结构化类与系统其余部分的交互点</p>
			<p>用例图：描述一组用例、参与者及它们之间的关系</p>
			<p>顺序图：是一种交互图 (Interaction Diagram)，交互图展现了一种交互，它由一组对象或参与者以及它们之间可能发送的消息构成</p>
			<p>通信图：也是一种交互图，它强调收发消息的对象或参与者的结构组织</p>
			<p>定时图：也是一种交互图，它强调消息跨越不同对象或参与者的实际时间，而不仅仅是关心消息的相对顺序</p>
			<p>状态图：描述一个状态机，它由状态、转移、事件和活动组成。状态图给出了对象的动态视图</p>
			<p>活动图：将进程或其他计算结构展示为计算内部一步步的控制流和数据流</p>
			<p>部署图：描述对运行时的处理节点及在其中生存的构件的配置</p>
			<p>制品图：描述计算机中一个系统的物理结构</p>
			<p>包图：描述由模型本身分解而成的组织单元，以及它们之间的依赖关系</p>
			<p>交互概览图：是活动图和顺序图的混合物</p>



软件实现	软件配置管理		软件配置管理通过标识产品的组成元素、管理和控制变更、验证、记录和报告配置信息，来控制产品的演进和完整性
	软件编码	程序设计语言	程序设计语言是人和计算机通信的最基本工具，编码之前的一项重要工作就是选择一种恰当的程序设计语言
		程序设计风格	程序设计风格包括 4 个方面：源程序文档化、数据说明、语句结构和输入 / 输出方法
		程序复杂性度量	定量度量程序复杂程度的方法很有价值，把程序的复杂度乘以适当的常数即可估算出软件中故障的数量及软件开发时的工作量
		编码效率	编码效率主要包括： (1) 程序效率 (2) 算法效率 (3) 存储效率 (4) I/O效率
	软件测试		软件测试的目的是验证软件是否满足软件开发合同或项目开发计划、系统 / 子系统设计文档、SRS、软件设计说明和软件产品说明等规定的软件质量要求
	软件部署与交付		软件部署与交付是软件生命周期中的一个重要环节，属于软件开发的后期活动，即通过配置、安装和激活等活动来保障软件制品的后续运行
	持续交付	持续交付是一系列开发实践方法，用来确保让代码能够快速、安全地部署到生产环境中。持续交付是一个完全自动化的过程，当业务开发完成的时候，可以做到一键部署	

部署交付	持续部署	持续部署方案	常用的持续部署方案有 Kubemetes+Docker 和 Matrix系统两种	
		部署原则	<ul style="list-style-type: none"><li>•部署包全部来自统一的存储库；</li><li>•所有的环境使用相同的部署方式；</li><li>•所有的环境使用相同的部署脚本；</li><li>•部署流程编排阶梯式晋级，即在部署过程中需要设置多个检查点，一旦发生问题可以有序地进行回滚操作；</li><li>•整体部署由运维人员执行；</li><li>•仅通过流水线改变生产环境，防止配置漂移；</li><li>•不可变服务器；</li><li>•部署方式采用蓝绿部署或金丝雀部署。</li></ul>	
		部署层次	<ul style="list-style-type: none"><li>•Build：跟传统的编译类似，将软件编译形成RPM包或者Jar包；</li><li>•Ship：则是将所需的第三方依赖和第三方插件安装到环境中；</li><li>•Run：就是在不同的地方启动整套环境。</li></ul>	
		不可变服务器	不可变服务器是一种部署模式，是指除了更新和安装补丁程序以外，不对服务器进行任何更改	
		蓝绿部署和金丝雀部署	<div>蓝绿部署：是指在部署的时候准备新旧两个部署版本，通过域名解析切换的方式将用户使用环境切换到新版本中，当出现问题的时候，可以快速地用户环境切回旧版本，并对新版本进行修复和调整</div> <div>金丝雀部署：是指当有新版本发布的时候，先让少量用户使用新版本，并且观察新版本是否存在问题。如果出现问题，就及时处理并重新发布；如果一切正常，就稳步地将新版本适配给所有的用户</div>	
过程管理	成熟度模型	部署与交付的新趋势	<div>工作职责和人员分工的转变</div> <div>大数据和云计算基础设施的普及进一步给部署带来新的飞跃</div> <div>研发运维的融合</div>	
		成熟度模型	CSMM 定义的软件过程能力成熟度模型旨在通过提升组织的软件开发能力帮助顾客提升软件的业务价值	
		成熟度等级	1 级：初始级	
			2 级：项目规范级	
			3 级：组织改进级	
4 级：量化提升级				
		5 级：创新引领级		



数据标准化	数据元		数据元是数据库、文件和数据交换的基本数据单元	
	数据元提取		数据元提取是数据元标准化的一项重要内容，为了确保数据元具有科学性和互操作性，需要采用合理的数据元提取方法	
	数据元标准化			描述
				开展业务流程分析与信息建模
				借助于信息模型，提取数据元，并按照一定的规则规范其属性
		数据元标准	基本过程	对于代码型的数据元，编制其值域，即代码表
				与现有的国家标准或行业标准进行协调
				发布实施数据元标准并建立相应的动态维护管理机制
	数据模式标准化			
	数据模式的描述方式主要有图描述方法和数据字典方法。图描述方法常用的有 IDEFIX 方法和 UML 图，主要用来描述数据集中的实体和实体之间的相互关系；数据字典形式用来描述模型中的数据集、单个实体、属性的摘要信息			
	数据分类与编码标准化			
	数据分类与编码标准化是简化信息交换、实现信息处理和信息资源共享的重要前提，是建立各种信息管理系统的重要技术基础和信息保障依据			
数据标准化管理	确定数据需求		本阶段将产生数据需求及相关的元数据、域值等文件	
	制定数据标准		本阶段要处理“确定数据需求”阶段提出的数据需求	
	批准数据标准		本阶段的数据管理机构对提交的数据标准建议、现行数据标准的修改或封存建议进行审查	
	实施数据标准		本阶段涉及在各信息系统中实施和改进已批准的数据标准	

信息系统工程

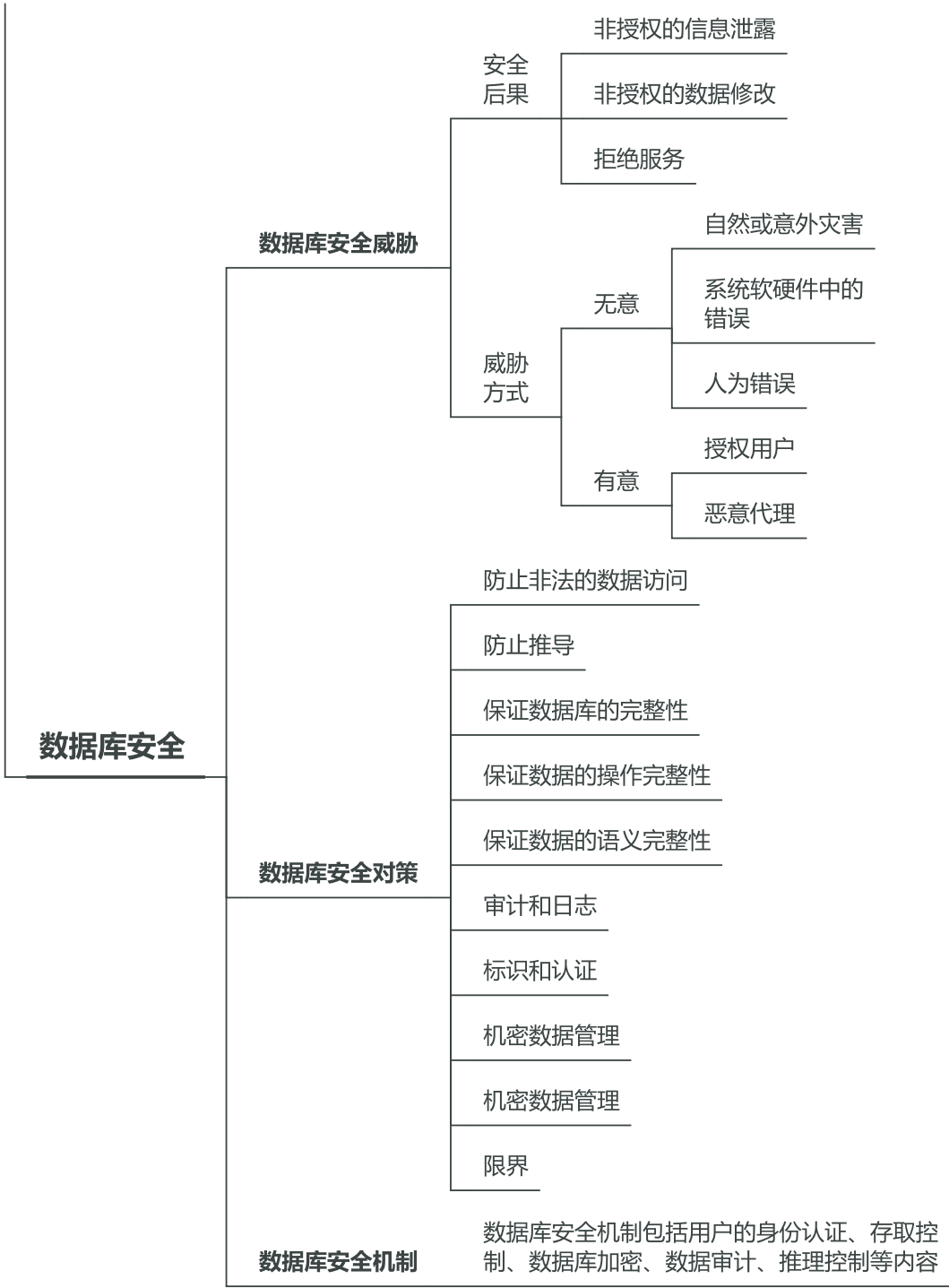
2.数据工程

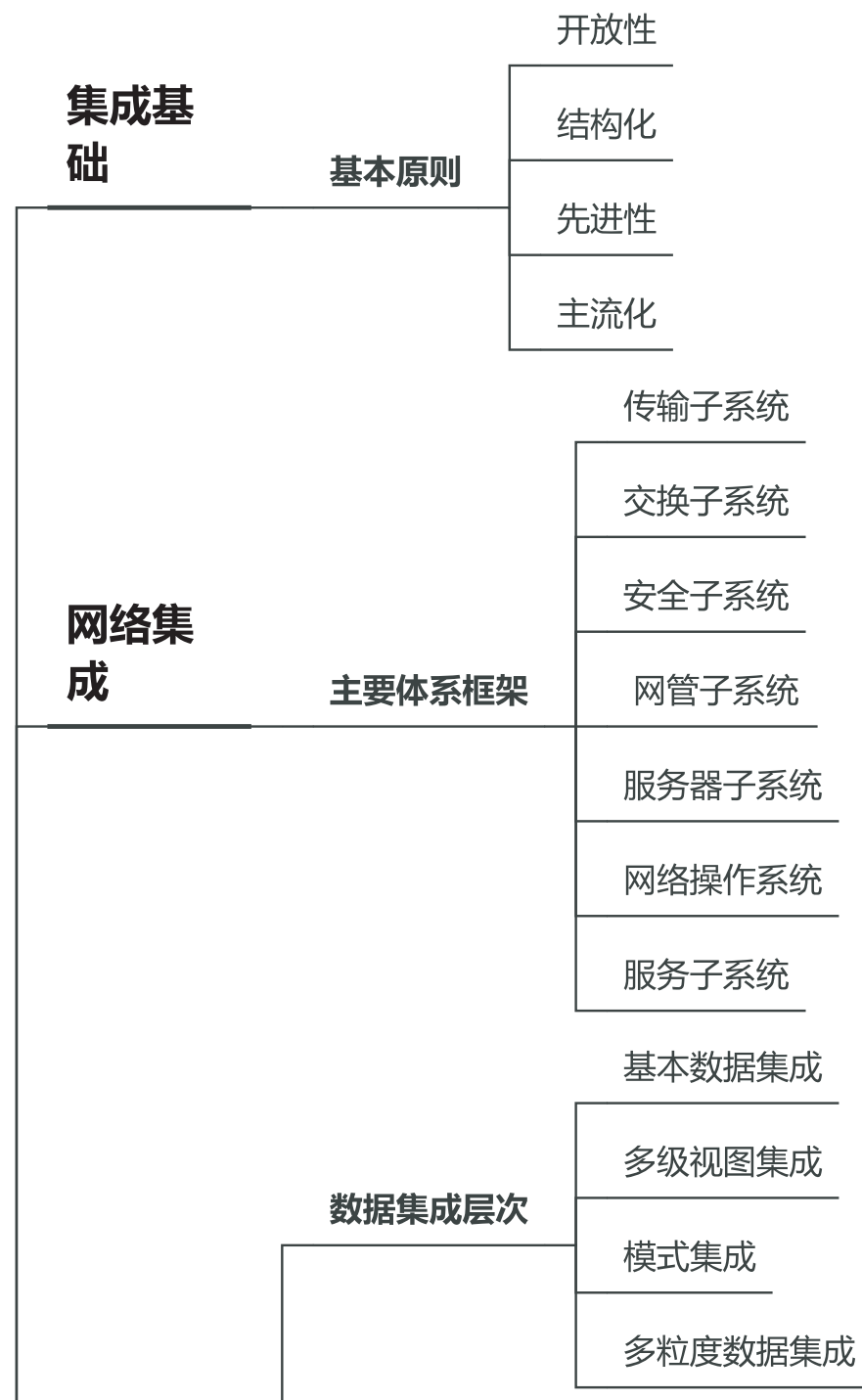
数据运维

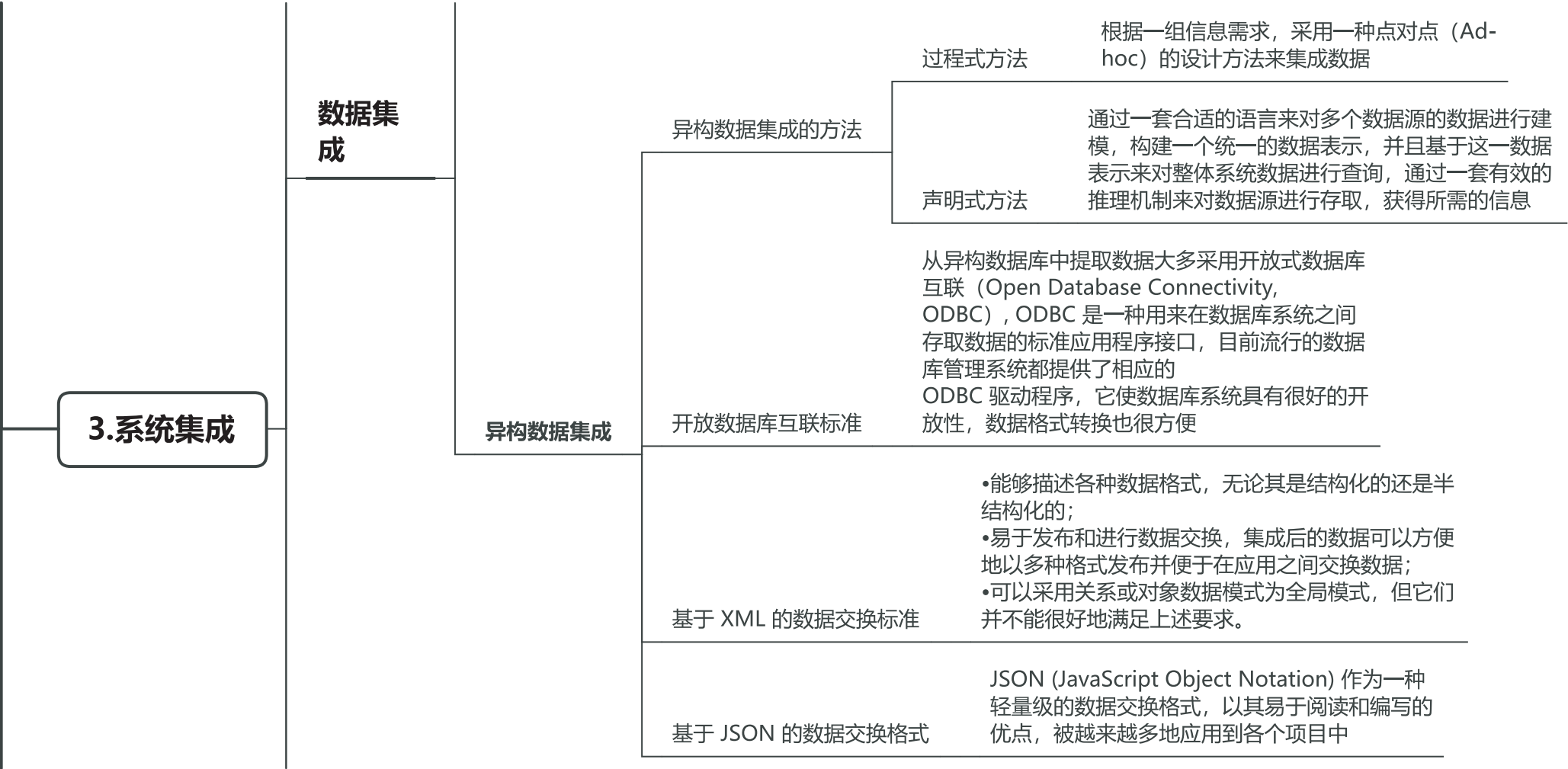
数据存储介质		数据存储首先要解决的是存储介质的问题。存储介质是数据存储的载体，是数据存储的基础。存储介质的类型主要有磁带、光盘和磁盘三种
数据存储	存储管理	资源调度管理
		存储资源管理
		负载均衡管理
		安全管理
数据备份是为了防止由于用户操作失误、系统故障等意外原因导致的数据丢失，而将整个应用系统的数据或一部分关键数据复制到其他存储介质上的过程		
数据容灾		
关注于保证用户数据的高可用性，在灾难发生时能够保证应用系统中的数据尽量少丢失或不丢失，使得应用系统能不间断地运行或尽快地恢复正常运行		
数据质量评价与控制	数据质量描述	数据质量可以通过数据质量元素来描述，数据质量元素分为数据质量定量元素和数据质量非定量元素
	数据质量评价过程	数据质量评价过程是产生和报告数据质量结果的一系列步骤
	数据质量评价方法	直接评价法：通过将数据与内部或外部的参照信息，如理论值等进行对比。确定数据质量
		间接评价法：利用数据相关信息，如数据只对数据源、采集方法等的描述推断或评估数据质量
	数据质量控制	前期控制：是在提交成果（即数据入库）之前对所获得的原始数据与完成的工作进行检查，进一步发现和改正错误
		过程控制：实施减少和消除误差和错误的实用技术和步骤，主要应用在建库过程中，用来对获得的数据在录入过程中进行属性的数据质量控制
		系统检测：在数据入库后进行系统检测，设计检测模板，利用检测程序进行系统自检
		精度评价：对入库属性数据用各种精度评价方法进行精度分析，为用户提供可靠的属性数据
	数据清理	数据分析：是指从数据中发现控制数据的一般规则，比如字段域、业务规则等，通过对数据的分析，定义出数据清理的规则，并选择合适的清理算法
		数据检测：是指根据预定义的清理规则及相关数据清理算法，检测数据是否正确，比如是否满足字段域、业务规则等，或检测记录是否重复
数据修正：是指手工或自动地修正检测到的错误数据或重复的记录		



数据开发利用	数据集成就是驻留在不同数据源中的数据进行整合，向用户提供统一的数据视图（一般称为全局模式），使得用户能以透明的方式访问数据	
	数据挖掘	数据挖掘是指从大量数据中提取或“挖掘”知识，即从大量的、不完全的、有噪声的、模糊的、随机的实际数据中，提取隐含在其中的、人们不知道的、却是潜在有用的知识
	数据服务	数据目录服务 用来快捷地发现和定位所需数据资源的一种检索服务，是实现数据共享的重要基础功能服务之一
		数据查询与浏览及下载服务 数据查询、浏览和下载是网上数据共享服务的重要方式，用户使用数据的方式有查询数据和下载数据两种
	数据可视化	数据分发服务 数据分发是指数据的生产者通过各种方式将数据传送到用户的过程
		一维数据可视化 一维数据可视化取决于数据大小和用户想用数据来处理什么任务
		二维数据可视化 最常见的二维数据可视化就是地理信息系统（Geographic Information System, GIS）
		三维数据可视化 物体通过三维可视化构成计算机模型，供操作及试验，以此预测真实物体的实际行为
		多维数据可视化 在可视化环境中，多维数据所描述事物的属性超过三维，为了实现可视化，往往需要降维
		时态数据可视化 以图形方式显示随着时间变化的数据，是可视化信息最常见、最有用的方式之一
		层次数据可视化 层次数据即树形数据，其数据内在结构特征为：每个节点都有一个父节点（根节点除外）
		网络数据可视化 网络数据指与任意数量的其他节点有关系的节点的数据
	信息检索	主要方法 <div>             (1) 全文检索              (2) 字段检索              (3) 基于内容的多媒体检索              (4) 数据挖掘           </div>
		常用技术 <div>             (1) 布尔逻辑检索技术              (2) 截词检索技术              (3) 临近检索技术              (4) 限定字段检索技术              (5) 限制检索技术           </div>







软件集成	CORBA	CORBA 是 OMG 进行标准化分布式对象计算的基础。CORBA 自动匹配许多公共网络任务，例如对象登记、定位、激活、多路请求、组帧和错误控制、参数编排和反编排、操作分配等
	COM	COM 具备了软件集成所需要的许多特征，包括面向对象、客户机/服务器、语言无关性、进程透明性和可重复性
	DCOM 与 COM+	DCOM 作为 COM 的扩展，不仅继承了 COM 优点，而且针对分布环境还提供了一些新的特性，如位置透明性、网络安全性、跨平台调用等
		COM+ 倡导了一种新的概念，它把 COM 组件软件提升到应用层而不再是底层的软件结构，通过操作系统的各种支持，使组件对象模型建立在应用层上，把所有组件的底层细节留给操作系统
	.NET	.NET 是基于一组开放的互联网协议，推出的一系列的产品、技术和服务
应用集成	J2EE	J2EE 架构是使用 Java 技术开发组织级应用的一种事实上的工业标准，它是 Java 技术不断适应和促进组织级应用过程中的产物
	应用集成或组织应用集成（EAI）是指将独立的软件应用连接起来，实现协同工作。借助应用集成，组织可以提高运营效率，实现 workflow 自动化，并增强不同部门和团队之间的协作	

