**第一章**

怎样理解整体大于部分之和？

系统要素协同配合时，将发挥出好的作用和效益，这就是整体大于部分之和。

当各个要素以有序、合理、优化的结构形成系统时，系统整体功能就会大于各部分功能之和。这是因为，系统整体是相互作用的各部分所构成的整体，整体的存在不仅依赖于部分的存在，而且依赖于各部分之间的相互性即内在联系，正是在部分之间的这种相互性和内在联系中产生了高效能的、有机的统一整体。最优化是系统方法所要达到的目标，它可以根据需要和可能为系统定量地确定出最优目标并运用最新技术手段和处理方法把整体系统逐阶分级，分成不同等级、层次结构，在动态中协调整体与部分的关系，使部分的功能和目标服从系统总体的最佳目标，以达到总体最佳。

**第二章**

信息系统的定义：信息系统就是输入数据，通过加工处理，产生信息的系统。在任何一个地方，只要有管理，就离不开信息系统。以计算机为基础的信息系统可以定义为：结合管理理论和方法，应用信息技术解决管理问题，为管理决策提供支持的系统。

信息系统的基本功能：具有数据的输入、传输、存储、处理、输出等基本功能。

信息系统的物理结构：集中式（单机系统） / 分布式（文件服务器F/S、客户机-服务器C/S、浏览器-服务器B/S）

集中式系统：是将软件、数据和主要外部设备集中在一套计算机系统中，采用集中计算模式。单机系统是典型的集中式系统。

分布式系统：通过计算机网络把不同地点的计算机硬件、软件、数据等资源联系在一起，共同服务于一个目标。实现不同地点的资源共享，是这种系统的主要特征。

缩写：

EDPS (electronic data processing system)数据处理系统

TPS（transaction processing system）事物处理系统

MIS（Management information system）管理信息系统

DSS(decision support system)决策支持系统 +group GDSS 群体决策支持系统

EIS（executive information system）主管信息系统

Data warehouse 数据仓库

Data mining 数据挖掘

Business intelligence BI 商务智能

ERP (enterprise resource planning)企业资源计划

EC (electronic commerce)电子商务

CRM(customer relationship management)客户关系管理

CMS（content management system）内容管理系统

SCM （supply chain management） 供应链管理

MRP (Material requirement planning )物料需求计划

MRPII （manufacturing resource planning ）制造资源计划

**描述未来信息系统的发展趋势？**

阿里云 分布式 人工智能平台 内容分发

ABCD AI Blockchain Cloud Data

**第三章**

**为什么说信息系统建设是复杂的社会过程？**

复杂性：技术手段复杂、内容复杂，目标多样，投资大，周期长，效益难以计算；环境复杂多变 涉及到组织内部各级机构、管理人员；思维方式不同。

社会过程:将信息系统建设与一般技术工程比较，信息系统建设的困难不仅来自技术方面，还来自企业内部环境。比如：体制、政策、法规、概念、技术等多种因素。

建设系统不单单是一个人的事，是关于整个团队的，要做许多的分析调研，工具开发，要通过很多建模工具来设计分析。

**信息系统的生命周期（信息系统建设的基本步骤）**

系统规划 任务：1.制定信息系统的发展战略2.制定信息系统的总体方案3.制定系统建设的资源分配计划 可划分为企业调查和分析、信息系统调查和分析制定方向、及评审建议 ；可行性可分为技术可行性、经济可行性、社会可行性（可行性报告、系统设计任务书）

分析 任务：做什么？系统分析员与用户在一起充分了解用户的要求，并把双方的理解用系统说明书表达出来，将业务流程与数据流进行建模，将业务流程图和数据流程图及数据字典写入系统说明书，系统说明书审核通过后将成为系统设计的依据，也是将来验收系统的依据（系统说明书）

设计 怎么做？实现系统说明书中所规定的功能。设计分为总体设计和详细设计总体设计时构建软件的总体框架 选择基本的逻辑架构（三层 表现层、业务逻辑层、数据访问层 五层 表现层、中介层、领域层、数据映射层、数据访问层 MVC架构）根据实际的技术条件、经济条件、和社会条件确定系统的实施方案即系统的物理模型。详细设计包括各分支技术的细节，将之前的设计整合为数据库设计和代码设计，首先数据库设计就是对当初数据字典当中的属性扩展成一张表，最后细致到每一句代码。为总体设计注入血肉，(系统设计说明书)

实施 任务：是实现系统设计阶段所提出的物理模型（实施进度和状态报告、系统测试分析报告）

运行与维护（记录系统运行的情况，定期整理成新需求建议书。为下一周期的系统规划做准备）

常用信息系统模型

业务流程图——不同职能部门业务活动分工和活动的过程

数据流图——数据的产生、处理、存储和去向的信息处理模型 系统分析阶段

程序流程图——程序完成顺序、分支、循环

（ER图）实体关系图——有价值的实体及其关系 系统设计阶段

组织结构图——部门及其从属关系

模块结构图——功能模块，调用关系

判定表、判定树——决策条件及其行动关系

UML——软件系统结构及行为

甘特图——项目任务以及其完成日期的项目计划

对系统进行概要描述：业务流程、数据流图

数据部分详细描述：数据字典

基于生命周期的开发方法：

瀑布（强调阶段的划分和阶段严格的顺序，一个新的项目不适合这个模型、适用于一些需求已明确并且变化较少的信息系统）

原型 用户积极参与 开发没有严密的阶段性 短期获得测试版本 。适应于需求含糊，用户不能详细的标示出准确的输入输出需求 设计方案不明确。

迭代 增量迭代过程好比搭积木，在每一步递增中，都可以把用户的经验结合到不断求精的下一个增量中，适用于项目开始，明确了需求的大部分，但是需求可能会发生变化；进化迭代过程好比滚雪球，每次迭代都是上一次迭代的演化和完善。

螺旋 风险驱动，要求用户参与阶段评价，开发人员具有丰富的风险评估经验，适用于 单位内部开发的大规模软件项目，风险是项目的主要制约因素

敏捷 基于迭代开发基础上

基于开发技术的开发方法：结构化（自顶向下分析与设计系统） 面向对象(把数据和对数据的操作紧密结合在一起：现实世界由对象组成；对象由属性和方法组成) 面向服务通常选取信息系统中某个关键功能作为原型（将程序进行封装）

**它们之间的联系 区别**

结构化

优点：从系统整体出发，强调整体优化的条件下“自上而下”地分析与设计，保证了系统地整体性和目标的一致性；遵循用户至上原则，严格区分系统开发的阶段性；每一阶段的工作成果是下一阶段地依据，整体性较好，便于系统开发地管理与控制，是其他系统开发（面向对象）的基础。

缺点：开发周期长

适用范围：规模较大、结构化程度较高

面向对象

优点：实现软件复用，简化程序设计，有利于维护，缩短开发周期

缺点：对开发人员要求高

使用范围：适用面很广

面向服务

优点：松耦合、自描述、可重用，可以将已经有的服务进行重新重组后可以快速建立新的业务流程。是以结构化和面向对象为基础。

系统开发项目管理内容包括：进度管理和控制、经费、质量、文档、人员。

**什么是CASE？**

在系统开发生命周期各个阶段的帮助开发者提高工作质量和效率的一类软件。

工具：分析与设计（Visio）、编程工具(Eclipse, Visual Studio 2010)、测试工具、运行维护工具、项目管理工具。

**第四章**

系统规划步骤：企业调查和分析、信息系统调查和分析、制定方向、评审及建议。

**第五章**

**系统分析基本任务**

系统分析员与用户在一起，充分了解用户的要求，并把双方的理解用系统说明书表达出来。系统说明书审核通过之后，将成为系统设计的依据，也是将来验收系统的依据。

**第六章**

数据流程图 由 外部实体、数据处理、数据流、数据存储等基本成分组成。

数据流程图的注意事项P121

数据字典种类：数据流、数据存储、处理过程、外部实体。

业务规则的描述工具有结构化语言、判定树、判定表。P132

**第七章**

**什么是用例？**

在面向对象方法中，用例描述系统需求。

用例创始人雅各布森 认为用例是对于一组动作序列的描述，系统执行这些动作会对特定的参与者产生可观测的、有价值的结果。

代表人物对上述用例概念进行扩展，强调用例是各种系统受益人之间的一种行为契约。

用例功能：

对系统需求进行规范化的描述，用例模型是面向对象分析的关键输入。

用例图及其事件流描述集中体现了系统责任。

通过用例交互图可以严格的对照检查系统的每一项功能需求是否得到满足，需求都落实到哪些对象，以及这些对象是如何来解决的。

用例包括：用例名称、参与者、前/后置条件、主事件/备选事件流、非功能性需求、业务规则。

**第八章**

面向对象方法三个阶段 ：面向对象编程/设计/分析

概念

对象、类、封装、消息、继承和多态性、关系。

**第九章**

系统设计的内容：系统总体设计、详细设计、系统实施进度与计划的制定、系统设计说明书的编写。

系统分析与系统设计的区别 所侧重的内容不一样 没有严格的先后顺序，可以相互结合。

**第十章**

软件框架模式：分布式、管道和过滤器、分层、MVC模式、黑板模式。

三层架构：表现层 业务逻辑层 数据访问层

五层架构：表现层 控制层/中介层 领域层 数据映射层 数据访问层。

MVC架构：model模型 controller 控制器 view 视图

结构化设计方法

模块：模块是一段可以用名字调用的程序段。模块具有输入输出、逻辑功能、运行程序、内部数据四种属性。

结构图主要成分：模块、调用、数据、控制信息。

模块结构图和数据流程图的区别：

数据流程图 不关心细节结构，关心功能模块、数据。

模块结构图描述的更加细致，可以用菱形选择性的调用，循环调用。

模块间的耦合：耦合度表明模块之间相互依赖的程度。

影响模块耦合的因素：联系方式、来往信息的作用、数量。

模块的内聚：反映模块内部联系的紧密程度

内聚分为：偶然、逻辑、时间、步骤、通信、顺序、功能。

设计一个好的结构化程序应该考虑哪些因素：模块间的耦合（越松散越好）、

模块的内聚、作用范围和控制范围、模块的扇入与扇出。

P202 实例

数据流图导出结构图P208

GOF设计模式：

创建型 抽象工厂模式（abstract factory）生成器模式（builder） 工厂方法模式（factory method）

结构型 桥模式（bridge）、组合模式（composite）、装饰模式（decorator）

行为型 命令模式（command）、状态模式（state）、备忘录模式（memento）

**为什么这些 能总结出GOF模式，有什么意义？**

**第十一章**

输入数据的校验方法：重复校验、视觉校验、分批汇总校验、控制总数校验、数据类型校验、格式校验、逻辑校验、界限校验、记录计数校验、匹配校验、代码自身校验。

数据库的设计：规范化 p264

代码设计原则：唯一、可扩充性、实用性及系统、简单、规范、系统

代码的种类：顺序码、矩阵码、自检码、系统顺序码、层次码

代码检验方法：1.加权求和2.以模除和的余数3将余数或摸和余数之差作为校验码附加在原编码之后。

代码的类型：数字、字母、数字字母混合

**第十二章**

什么是好程序：能够工作、调试代价低、易于修改、易于维护、设计不复杂、效率高

系统测试：模块测试、联合测试、验收测试、系统测试

白箱测试：是根据软件不见内部控制结构，测试是否依据设计正确地执行

黑箱测试：是根据一个软件部件由外部界面所能观察到地功能效果，测试他是否能与其他部件正常的沟通。

测试用例设计方法：语句覆盖法、判断、条件、组合、路径、等价类划分、边界值测试

排错 ：试探、跟踪、对分查找法、归纳法、演绎法

**第十三章**

概念题：

维护的内容：应用程序的维护、数据文件的维护、代码的维护、机器设备的维护

维护的类型:改正性维护、适应性维护、完整性维护、预防性维护

**第十四章**

云计算的概念、定义P297