

考点	重要程度	占分	题型
4.1 设计过程	★★★	2~6	选择、填空、简答
4.2 设计原理	★★★★	4~8	填空、选择、简答
4.3 启发规则	★★★	4~6	填空、选择、简答
4.4 描绘软件结构的图形工具	★★★	2~6	选择、填空、画图
4.5 面向数据流设计方法	★★	2~4	选择、填空

4.1 设计过程

总体设计又称为概要设计或初步设计。

任务：

- 确定系统中每个程序由哪些模块组成以及这些模块相互间的关系。
- 划分出物理元素。包括程序、文件、数据库、文档等。

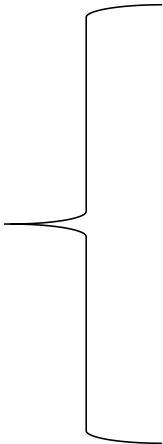


视频讲解更清晰
仅3小时

4.1设计过程

设计过程包括系统设计阶段和结构设计阶段

系统设计阶段



- 1.设想供选择的方案: 数据流图出发, 将处理分组抛弃行不通分组。
- 2.选取合理的方案:上一步方案选取低、中、高成本三种方案。
- 3.推荐最佳方案: 推荐最佳方案, 制定详细实现计划

4.1设计过程

设计过程包括系统设计阶段和结构设计阶段

结构设计阶段

- 4.功能分解：对数据流图进一步细化，进行功能分解。可以用IPO图等工具描述细化后每个处理的算法。
- 5.设计软件结构：层次图或结构图描绘软件结构。或数据流图导出软件结构。
- 6.设计数据库
- 7.制定测试计划
- 8.书写文档
- 9.审查和复审

4.2设计原理

- 模块化

模块：能够单独命名，由边界元素限定的程序元素的序列，是构成程序的基本构件。

模块化：把程序划分成独立命名且可独立访问的模块，每个模块完成一个子功能，把这些模块集成起来构成一个整体，可以完成指定的功能满足用户的需求。

- 抽象

抽出事务的本质特性而暂时不考虑它们的细节。

- 逐步求精

逐步揭露出底层细节。

Miller法则：注意力集中在 (7 ± 2) 上

4.2设计原理

- 信息隐藏与局部化

信息隐藏： 指一个模块内包含的信息对于不需要这些信息的模块来说，是不能访问的。主要是指模块的实现细节。

局部化： 指把一些关系密切的软件元素物理地放得彼此靠近，它有助于实现信息隐藏。

- 模块独立

模块独立性： 是模块化、抽象、信息隐蔽和局部化概念的直接结果。

模块独立是好设计的关键，设计是决定软件质量的关键环节。

度量标准： 耦合、内聚

4.2设计原理

一、耦合

是对一个软件结构内不同**模块之间**互连程序的度量。

耦合强度取决于模块接口的复杂程度、通过接口的数据等

耦合性越高，模块独立性越弱。

耦合分类(程度从低->高):

无直接耦合=》数据耦合=》标记耦合(特征耦合)=》控制耦合
=》外部耦合=》公共耦合

4.2设计原理

二、内聚

是用来度量一个**模块内部**各个元素彼此结合的紧密程度的。

内聚分类(程度从低->高):

偶然内聚=》逻辑内聚=》时间内聚=》过程内聚=》通信内聚=》

顺序内聚=》功能内聚

同其它模块强耦合的模块意味着弱内聚；强内聚模块意味着与其它模块间松散耦合

软件设计目标:高内聚、低耦合

4.3启发规则

- 改进软件结构提高模块独立性
- 模块规模应该适中
- 深度、宽度、扇入和扇出应适当

深度：表示软件结构中控制的层数。

宽度：软件结构内同一个层次上的模块总数的最大值。

扇出：一个模块直接控制（调用）的模块数目，扇出过大意味着模块过分复杂。一般一个设计的好的典型系统的平均扇出是3或4，扇出的上限是5到9。

扇入：指有多少上级模块调用它，扇入大说明上级模块共享该模块的数目多。

好的软件结构**顶层扇出比较高**，**中层扇出比较少**，底层扇入到公共的实用模块中，即**底层模块有高扇入**。

4.3启发规则

- 模块的作用域应该在控制域之内

作用域：指受该模块内一个判定影响的所有模块的集合。

控制域：是这个模块本身以及所有直接或间接从属于它的模块的集合。

- 力争降低模块接口的复杂程度
- 设计单入口单出口的模块
- 模块功能应该可以预测

4.4描绘软件结构的图形工具

一、层次图与IPO图

层次图:用方框和连线表示,连线表示上下层的调用关系。



4.4描绘软件结构的图形工具

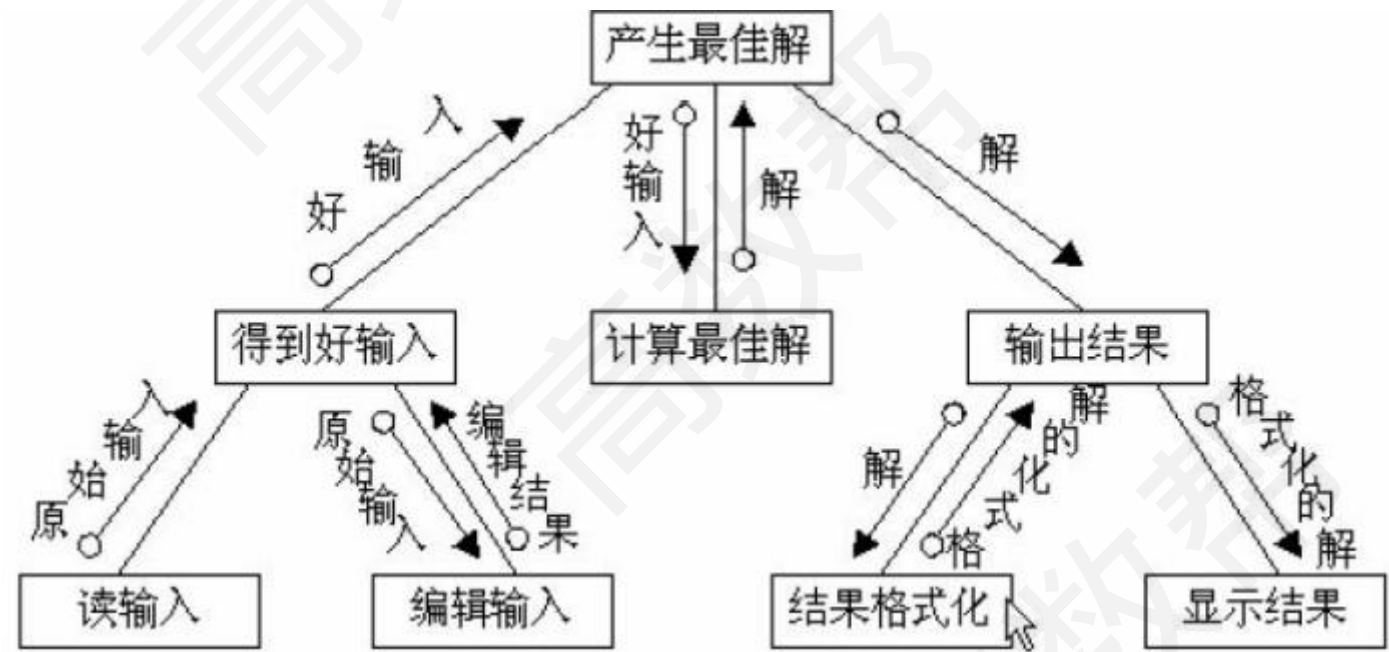
一、层次图与IPO图

HIPO图：层次图加编号



二、结构图

不仅描述调用关系，还描述传递的信息和调用方式。



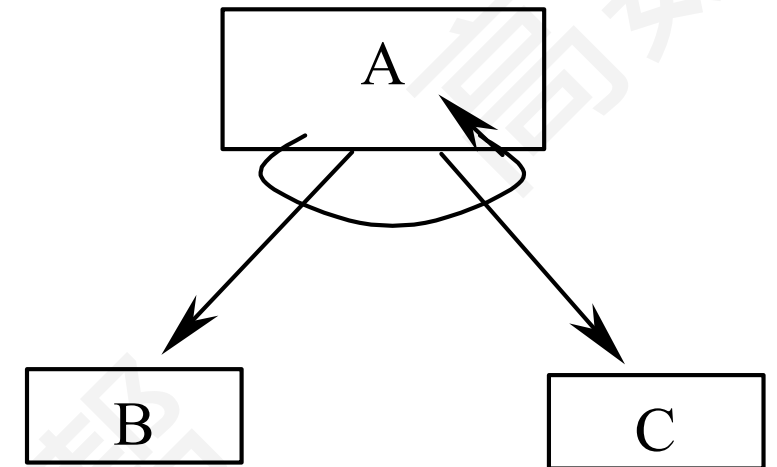
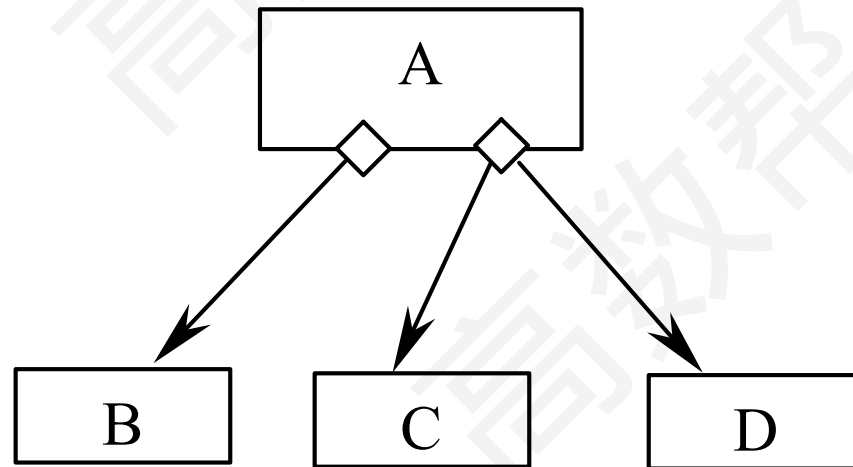
箭头代表调用过程传递的信息。尾部空心代表**数据**，实心代表**控制信息**。

4.4描绘软件结构的图形工具

二、结构图

模块调用分类

- 1.简单调用
- 2.循环调用
- 3.选择调用

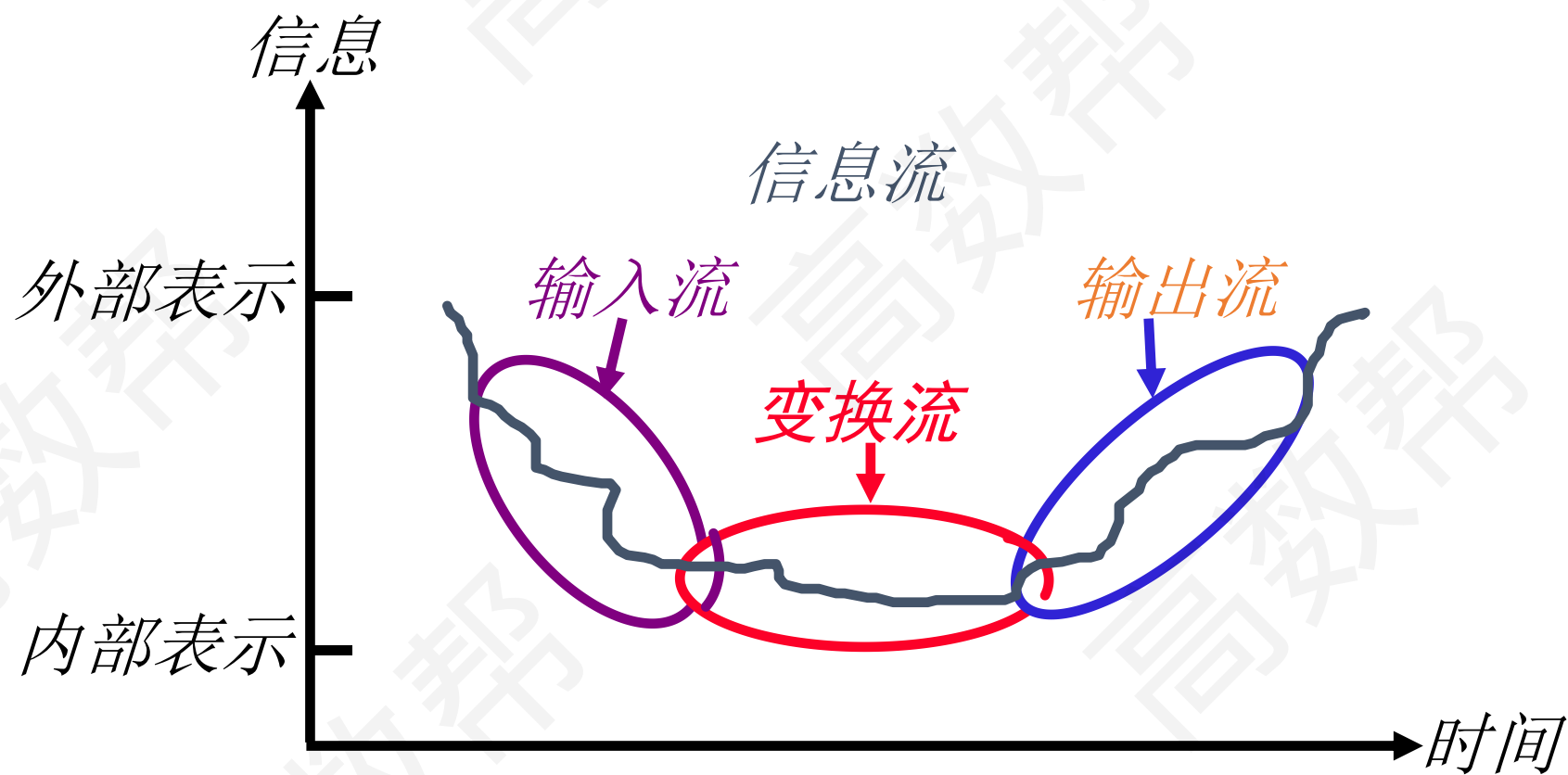


4.5面向数据流设计方法

面向数据流设计方法也称为结构化设计方法（SD）。

数据流图分类

(1) **变换流**：由输入、变换中心和输出三部分组成。

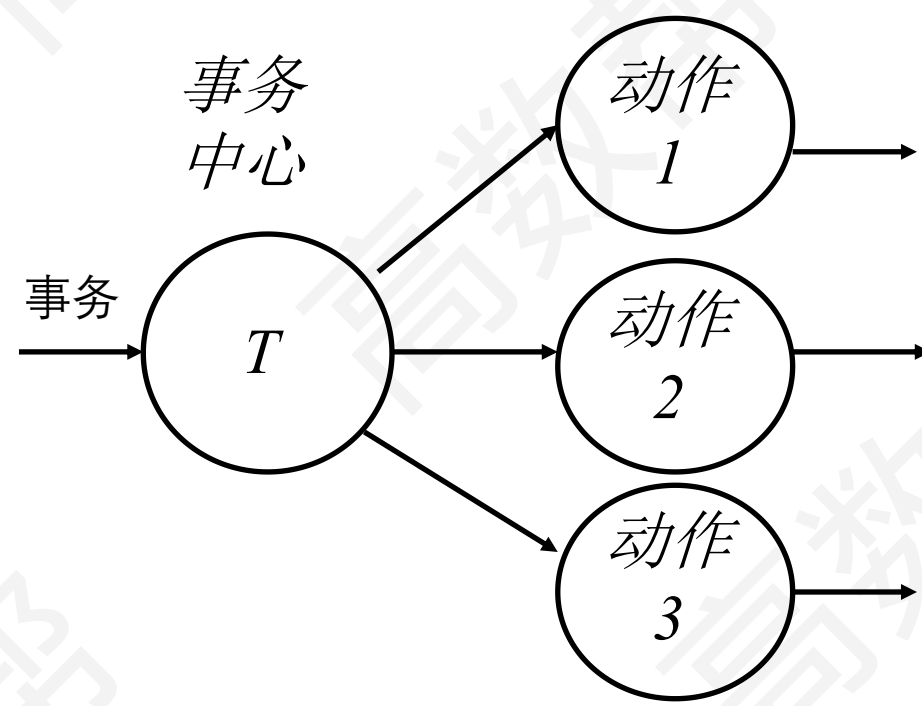


4.5面向数据流设计方法

面向数据流设计方法也称为结构化设计方法（SD）。

数据流图分类

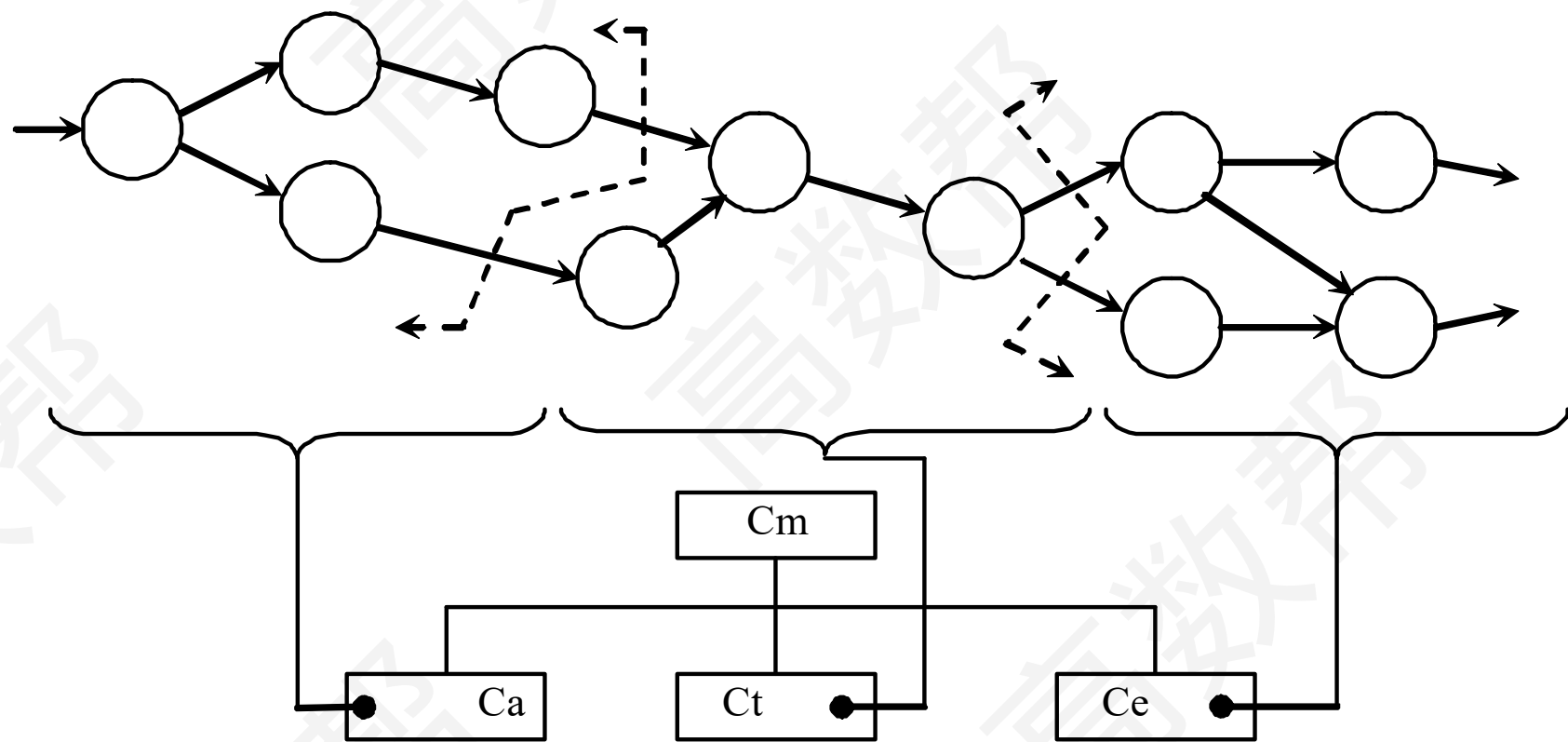
(2) **事务流**：在多种事务中选择一个执行。



4.5面向数据流设计方法

变换分析:

把具有变换流特点的数据流图映射成软件结构。



总结

掌握

- 设计过程
- 设计原理
- 启发规则
- 结构设计图形工具

了解

结构化设计方法



视频讲解更清晰
仅3小时