**软件设计师高频考点  
上午题章节  
第一章 计算机组成与体系结构 2  
第二章 操作系统 3  
第三章 数据库系统 4  
第四章 计算机网络 5  
第五章 信息安全 6  
第六章 软件工程 7  
第七章 项目管理 8  
第八章 面向对象技术 9  
第九章 数据结构与算法基础 10  
第十章 程序设计语言 11  
第十一章 法律法规与标准化 11**

**下午题章节  
 第一章 数据流图 12  
 第二章 数据库设计 14  
 第三章 UML建模 15  
 第四章 数据结构与算法应用 17  
 第五章 面向对象程序设计 18  
  
  
  
 第一章 计算机组成与体系结构**

**高频考点介绍：**1、CPU的组成（运算器与控制器）  
2、流水线技术的计算  
3、CISC和RISC  
4、主存编址计算  
5、输入/输出技术  
6、校验码（海明校验码，奇偶校验码，循环冗余校验码）  
5、层次化存储体系结构与Cache  
7、可靠性与可用性  
8、进制转换与码制（原、反、补、移码）

**第二章 操作系统**

**高频考点介绍：**  
1、操作系统概述及其分类  
2、线程的概念与进程的状态（三态模型）  
3、信号量取值范围计算  
4、PV操作及其前驱图  
5、死锁及其资源数计算与进程资源图  
6、页式存储与页面置换，以及段页式存储  
7、磁盘管理和I/O管理软件  
8、文件系统相关概念和树形目录结构（绝对路径和相对路径）  
9、位示图计算和索引文件的计算。

**第三章 数据库系统**

**高频考点介绍：**  
1、分布式数据库的考查  
2、数据库三级模式两级映射  
3、数据库设计过程及概念结构设计过程  
4、ER模型  
5、关系代数  
6、规范化理论以及范式（1NF，2NF，3NF，BCNF）判断  
7、保持函数依赖与模式分解

8、查询语句（普通查询、分组查询、权限控制、其他SQL语句）  
9、事务的特性和封锁协议

**第四章 计算机网络**

**高频考点介绍：**  
1、开放系统互联参考模型  
2、常见的TCP/IP协议基础  
3、TCP与UDP  
4、网络故障以及常用命令  
5、IP地址的分类与特殊的IP地址  
6、子网划分与路由汇聚  
7、IPV4与IPV6  
8、网络规划设计与网络接入技术以及路由配置  
9、URL与HTML语言

**第五章 信息安全**

**高频考点介绍：**  
1、对称加密与非对称加密技术  
2、信息摘要与数字签名以及数字证书  
3、网络安全协议与网络攻击  
4、计算机病毒与木马以及分类  
5、防火墙技术以及其他网络安全控制技术  
6、系统安全分级

**第六章 软件工程**

**高频考点介绍：**  
1、开发模型与软件开发方法  
2、需求分析相关概念以及需求分类  
3、数据流图与数据字典  
4、概要设计与详细设计  
5、模块设计原则的内聚和耦合性  
6、人机界面设计与架构设计  
7、软件测试的黑盒测试与白盒测试以及其他测试

8、McCabe复杂度计算  
9、软件维护类型与软件过程改进  
10、软件质量特性与软件文档

**第七章 项目管理**

**高频考点介绍：**  
1、Gant图与Pert图  
2、风险管理  
3、配置管理  
4、沟通管理  
5、软件项目估算

**第八章 面向对象技术**

**高频考点介绍：**  
1、面向对象开发各阶段划分及其任务  
2、面向对象的基本概念  
3、面向对象设计原则  
4、UML图的定义与关系  
5、UML图的图示  
6、设计模式的判断及其适用场景

**第九章 数据结构与算法**

**高频考点介绍：**  
1、数组与矩阵  
2、顺序表与链表  
3、队列与栈、字符串  
4、树与二叉树的特性  
5、二叉树的遍历  
6、二叉树的排序树与最优二叉树   
7、图的定义及其存储

8、图的遍历以及拓扑排序  
9、二分查找与排序、散列表  
10、时间复杂度与空间复杂度  
11、四种常见算法（动态规划法、贪心算法、分治法、回溯法）

**第十章 程序设计语言**

**高频考点介绍：**  
1、编译与解释以及编译工作过程  
2、中间代码与错误管理  
3、文法与正规式、有限自动机  
4、后缀表达式  
5、传值与传址  
6、多种程序设计语言的数据成分与控制成分以及多种程序语言

**第十一章 法律法规与标准化**

**高频考点介绍：**  
1、保护期限  
2、保护范围与对象  
3、知识产权人确定  
4、侵权判断

**第一章 数据流图**

**一、数据流图的基本概念**  
1、数据流：一组固定成分的数据组成，表示数据流向（数据流必须与加工有关）

2、加工：描述了输入数据流到输出数据流之间的变换  
3、数据存储：用来表示暂时存储的数据  
4、外部实体：指存在于软件系统外的人员或组织  
  
**二、加工逻辑的描述**  
1、结构化语言  
2、判定表  
3、判定树  
  
**三、数据流图平衡原则**  
1、父图与子图之间的平衡   
指任何一张DFD子图边界上的输入/输出数据流必须与其父图对应加工的输入/输出保持一致。  
2、子图内平衡  
（1）加工只有输入没有输出，称之为“黑洞”  
（2）加工只有输出没有输入，称之为“奇迹”  
（3）加工中输入不足以产生输出，称之为“灰洞”

**四、做题技巧**

1、补充实体  
实体可能是：  
（1）人物角色：如客户、管理员、主管、经理、老师、学生  
（2）组织机构：如银行、供应商、募捐机构

(3) 外部系统：如银行系统、工资系统、后台数据库  
2、补充存储  
存储的文字方面特征：“文件”，“表”，“库”，“清单”，“档案”  
3、补充数据流  
根据数据流图平衡原则和题干说明  
4、补充加工名  
根据题干说明，由“动词+名词”构成

**第二章 数据库设计**

**一、数据库设计过程**  
1、需求分析：数据流图、数据字典、需求说明书

2、概念结构设计：ER模型  
3、逻辑结构设计：关系模式  
4、物理设计：DBMS特性，硬件、OS特性  
  
**二、E-R图向关系模型的转换**  
1、一对一联系（可以将主键归并到任意一端，或者把两者主键都归并到新的联系）  
2、一对多联系（可以将主键归并到多端，或者把两者主键都归并到新的联系）  
3、一对多联系（只能把两者主键都归并到新的联系）  
  
**三、答题技巧**  
1、联系和联系判断：根据题干找出关键信息，判断实体联系的比值。  
2、关系模式属性判断：根据题干找出遗失的属性，或根据ER转换为关系模型时属性的增添。  
3、需要掌握的基本知识：E-R图，联系类型判断，E-R图转换为关系模式，主键、外键判断，规范化理论。

**第三章 UML建模**

**一、用例图**  
描述了一组用例，参与者及题它们之间的关系  
关系包括：  
（1）包含关系：可以从两个及以上用例中提取公共行为时，使用包含关系，用include表示  
（2）扩展关系：一个用例明显地混合了两种或以上的不同场景，根据情况发生分支，可以将该用例分为一个基本用例和多个扩展用例，用extend表示  
（3）泛化关系：父子之间的关系，一般与特殊的关系

**二、类图与对象图**类图：描述一组类、接口、协作和它们之间的关系。  
对象图：对象图描述一组对象及它们之间的关系。  
  
**三、顺序图**顺序图是一种交互图，交互图展现了一种交互，由一组对象或参与者及它们之间可能发送的消息构成。顺序图强调消息的时间次序的交互。  
  
**四、活动图**  
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。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。

**五、状态图**  
状态图描述了一个状态机，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。它对于接口、类或协作的行为建模尤为重要，强调事件导致的对象行为。  
  
**六、通信图**  
通信图是一种交互图，交互图展现了一种交互，由一组对象或参与者及它们之间可能发送的消息构成。通信图强调的是对象之间的组织结构。  
  
**七、构件图**  
构件图描述了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。  
  
**八、部署图**  
部署图描述了对运行时的处理节点及在其中生存的构建的配置。强调了软件和硬件之间部署关系。

**第四章 数据结构与算法应用**

**一、常见算法特征总结**  
1、分治法  
特征：把一个问题拆分成多个小规模的相同子问题，一般用递归解决。  
经典问题：斐波那契数列、归并排序、快速排序、矩阵乘法、二分搜索、汉诺塔、大整数乘法。  
2、动态规划法（用于求最优解）--“最优子结构”和递归式  
特征：划分子问题，并把子问题结果使用数组存储，利用查询子问题结果构造最终问题结果。  
经典问题：斐波那契数列、矩阵乘法、背包问题、LCS最长公共子序列  
3、回溯法  
特征：系统的搜索一个问题的所有解或任一解  
经典问题：N皇后问题、迷宫、背包问题  
4、贪心法（一般用于求满意解）  
特征：局部最优，但整体不见得最优。每步有明确的，既定的策略。  
经典问题：背包问题（如装箱）、多机调度、找零钱问题

**第五章 面向对象程序设计**

**一、代码填空技巧（针对JAVA）**  
1、关键字填空  
经常出现关键字填空，比如abstract、extends、implements等，注意不要出现拼写错误。  
2、缺失方法填空  
（1）注意父类与子类、接口与实现类之间的统一，经常出现对于方法的缺失，可以根据上下文进行填写，注意接口方法无方法体，其他返回值类型、参数列表应该保持一致；  
（2）注意构造函数的写法；  
（3）如果JAVA自带的一些类型的方法，需要靠平常的积累，建议多做真题进行积累。  
3、实例化填写  
注意实例化的参数选择  
4、注意函数调用(参数列表、参数类型、对应参数能够调用的方法)  
5、在软件设计师考试中，程序设计题最难的地方在于与设计模式的结合，建议多做真题进行积累。