[第1章 计算机组成与体系结构 1](#_Toc74672536)

[1 考情分析 1](#_Toc74672537)

[1.1 本章重点 1](#_Toc74672538)

[2 考点精讲 2](#_Toc74672539)

[2.1 数据的表示（★★★★） 2](#_Toc74672540)

[2.2 运算器与控制器（★★★★） 3](#_Toc74672541)

[2.3 Flynn分类法（★★） 4](#_Toc74672542)

[2.4 CISC与RISC（★★） 4](#_Toc74672543)

[2.5 流水线技术（★★★★） 5](#_Toc74672544)

[2.6 存储系统（★★★★） 6](#_Toc74672545)

[2.7 总线系统（★） 8](#_Toc74672546)

[2.8 可靠性（★） 8](#_Toc74672547)

[2.9 校验码（★★★） 9](#_Toc74672548)

[2.10 计算机性能指标（★） 9](#_Toc74672549)

[3 章节问答 10](#_Toc74672550)

[第2章 操作系统 11](#_Toc74672551)

[1 考情分析 11](#_Toc74672552)

[1.1 本章重点 11](#_Toc74672553)

[2 考点精讲 11](#_Toc74672554)

[2.1 进程的状态（★★） 13](#_Toc74672555)

[2.2 前趋图（★★★） 14](#_Toc74672556)

[2.3 信号量与PV操作（★★★★） 14](#_Toc74672557)

[2.4 死锁及银行家算法（★★★★） 15](#_Toc74672558)

[2.5 段页式存储（★★★★） 16](#_Toc74672559)

[2.6 页面置换算法（★） 16](#_Toc74672560)

[2.7 磁盘管理（★★） 16](#_Toc74672561)

[2.8 绝对路径与相对路径（★★★） 17](#_Toc74672562)

[2.9 索引文件（★★） 17](#_Toc74672563)

[2.10 位示图（★★） 18](#_Toc74672564)

[2.11 I/O设备管理（★） 18](#_Toc74672565)

[3 章节问答 19](#_Toc74672566)

[第3章 数据库系统 22](#_Toc74672567)

[1 考情分析 22](#_Toc74672568)

[1.1 本章重点 22](#_Toc74672569)

[2 考点精讲 22](#_Toc74672570)

[2.1 数据库模式（★★） 22](#_Toc74672571)

[2.2 数据库设计过程（★★） 23](#_Toc74672572)

[2.3 ER模型（★★★★★） 23](#_Toc74672573)

[2.4 关系代数（★★★） 26](#_Toc74672574)

[2.5 规范化理论（★★★★★） 28](#_Toc74672575)

[2.6 SQL语言（★★★★） 29](#_Toc74672576)

[2.7 并发控制（★★） 30](#_Toc74672577)

[2.8 数据库完整性约束（★） 31](#_Toc74672578)

[2.9 数据库扩展知识（★） 31](#_Toc74672579)

[2.10 【软件设计】数据库设计题解题技巧 32](#_Toc74672580)

[3 章节问答 33](#_Toc74672581)

[第4章、计算机网络与信息安全 35](#_Toc74672582)

[1 考情分析 35](#_Toc74672583)

[1.1 本章重点 35](#_Toc74672584)

[2 考点精讲 36](#_Toc74672585)

[2.1 开放系统互连参考模型OSI（★★） 36](#_Toc74672586)

[2.2 TCP/IP协议族（★★★★） 36](#_Toc74672587)

[2.3 IP地址与子网划分（★★★★★） 37](#_Toc74672588)

[2.4 网络规划与设计（★） 38](#_Toc74672589)

[2.5 网络接入技术（★★） 39](#_Toc74672590)

[2.6 HTML语言（★★） 40](#_Toc74672591)

[2.7 URL（★） 40](#_Toc74672592)

[2.8 对称加密与非对称加密（★★★） 41](#_Toc74672593)

[2.9 信息摘要与数字签名（★★） 41](#_Toc74672594)

[2.10 数字证书（★★） 42](#_Toc74672595)

[2.11 网络安全协议（★★★） 42](#_Toc74672596)

[2.12 防火墙技术与网络攻击（★★★） 43](#_Toc74672597)

[2.13 计算机病毒与木马（★） 45](#_Toc74672598)

[3 章节问答 45](#_Toc74672599)

[第5章、系统开发基础 47](#_Toc74672600)

[1 考情分析 47](#_Toc74672601)

[1.1 本章重点 47](#_Toc74672602)

[2 知识点详情 48](#_Toc74672603)

[2.1 开发模型（★★★★★） 48](#_Toc74672604)

[2.2 软件开发方法（★） 49](#_Toc74672605)

[2.3 需求分析（★） 49](#_Toc74672606)

[2.4 软件设计（★★） 49](#_Toc74672607)

[2.5 软件测试（★★★★） 51](#_Toc74672608)

[2.6 McCabe复杂度计算（★★★） 52](#_Toc74672609)

[2.7 软件维护类型（★★★★） 53](#_Toc74672610)

[2.8 软件质量保证（★） 53](#_Toc74672611)

[2.9 软件过程改进（★★） 53](#_Toc74672612)

[2.10 Gantt图与Pert图（★★★★） 54](#_Toc74672613)

[2.11 风险管理（★★★） 56](#_Toc74672614)

[2.12 沟通管理（★） 56](#_Toc74672615)

[2.13 【软件设计】数据流图解题技巧 56](#_Toc74672616)

[3 章节问答 58](#_Toc74672617)

[第6章、面向对象 60](#_Toc74672618)

[1 考情分析 60](#_Toc74672619)

[1.1 本章重点 60](#_Toc74672620)

[2 知识点详情 60](#_Toc74672621)

[2.1 面向对象的概念（★★★★★） 60](#_Toc74672622)

[2.2 UML（★★★★★） 61](#_Toc74672623)

[2.3 【软件设计】UML建模解题技巧 66](#_Toc74672624)

[2.4 设计模式（★★★★★） 67](#_Toc74672625)

[2.5 【软件设计】面向对象程序设计（JAVA）解题技巧 70](#_Toc74672626)

[3 章节问答 73](#_Toc74672627)

[第7章 数据结构 75](#_Toc74672628)

[1 考情分析 75](#_Toc74672629)

[1.1 本章重点 75](#_Toc74672630)

[2 知识点详情 75](#_Toc74672631)

[2.1 数组与矩阵（★★） 75](#_Toc74672632)

[2.2 线性结构（★★★★★） 76](#_Toc74672633)

[2.3 广义表（★★） 78](#_Toc74672634)

[2.4 树与二叉树（★★★★★） 78](#_Toc74672635)

[2.5 图（★★） 80](#_Toc74672636)

[3 章节问答 81](#_Toc74672637)

[第8章 算法基础 83](#_Toc74672638)

[1 考情分析 83](#_Toc74672639)

[1.1 本章重点 83](#_Toc74672640)

[2 知识点详情 83](#_Toc74672641)

[2.1 算法基础及常见算法（★★★★★） 83](#_Toc74672642)

[2.2 时间复杂度与空间复杂度（★★★★★） 84](#_Toc74672643)

[2.3查找（★★★★★） 85](#_Toc74672644)

[2.4 排序（★★★★★） 86](#_Toc74672645)

[第9章、程序设计语言与语言处理程序基础 91](#_Toc74672646)

[1 考情分析 91](#_Toc74672647)

[1.1 本章重点 91](#_Toc74672648)

[2 知识点详情 91](#_Toc74672649)

[2.1 编译与解释（★★★） 91](#_Toc74672650)

[2.2 文法（★★） 93](#_Toc74672651)

[2.3 正规式（★★★★） 94](#_Toc74672652)

[2.4 有限自动机（★★） 95](#_Toc74672653)

[2.5 后缀表达式（★★★） 95](#_Toc74672654)

[2.6 传值与传址（★★★★） 95](#_Toc74672655)

[2.7 多种程序语言特点（★★★） 96](#_Toc74672656)

[第10章知识产权与标准化 98](#_Toc74672657)

[1 考情分析 98](#_Toc74672658)

[1.1 本章重点 98](#_Toc74672659)

[2 知识点详情 98](#_Toc74672660)

[2.1 保护对象和保护期限（★★） 98](#_Toc74672661)

[2.2 知识产权人确定（★★★） 99](#_Toc74672662)

[2.3 侵权判断（★★★★） 100](#_Toc74672663)

[2.4 标准的分类（★） 101](#_Toc74672664)

[2.5 标准代号的识别（★） 101](#_Toc74672665)

# 

# 第1章 计算机组成与体系结构

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）掌握数据的进制转换（十进制、二进制、十六进制互转）。

（2）掌握原码/反码/补码/移码的转换规则，了解各种码制的表示范围和表示个数（注意补码和移码的人为定义）。

（3）掌握浮点数的格式，及各部分含义，了解浮点数运算的逻辑过程。

（4）掌握CPU各子部件的功能及其分类。

（5）了解Flynn分类法的分类及各类的典型系统。

（6）了解指令的基本分类、寻址方式分类。

（7）了解CISC与RISC的特点区别。

（8）掌握流水线相关的参数计算（流水线周期、流水线建立时间、流水线执行时间、流水线执行时间实践公、流水线吞吐率、流水线最大吞吐率等）。

（9）了解计算机分层存储结构相关概念（效率、容量、成本对比，局部性原理支撑）。

（10）了解局部性原理，熟悉Cache的相关概念和特点，区分Cache三种映射方式。

（11）掌握主存计算题型，掌握相关参数的计算（存储单元个数，存储总容量，单位芯片容量，芯片个数）。

（12）了解总线的分类——数据总线、控制总线、地址总线。

（13）掌握系统可靠性计算（串联系统、并联系统、混联系统）。

（14）了解奇偶校验，了解CRC校验（利用模二除计算），掌握海明校验位计算，区分三种校验方式对检错和纠错的特点。

（15）了解常见的计算机性能指标。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 数据的表示（★★★★） | 进制的转换 |
| 2 | 原码/反码/补码/移码表示 |
| 3 | 浮点数的表示 |
| 1 | 运算器与控制器（★★★★） | 运算器与控制器 |
| 1 | Flynn分类法（★★） | Flynn分类法 |
| 1 | CISC与RISC（★★） | CISC与RISC |
| 1 | 流水线技术（★★★★） | 流水线技术 |
| 1 | 存储系统（★★★★） | 分级存储体系 |
| 2 | Cache |
| 3 | 内存 |
| 1 | 总线系统（★） | 总线系统 |
| 1 | 可靠性（★） | 可靠性 |
| 1 | 校验码（★★★） | 校验码 |
| 1 | 计算机性能指标（★） | 性能指标 |

## 2 考点精讲

### 2.1 数据的表示（★★★★）

#### 2.1.1 进制的转换

【考法分析】

本考点的基本考法是与内存地址计算、IP地址计算结合考查。

【要点分析】

1、十进制转R进制（短除法）；

2、R进制转十进制（按权展开法）；

3、二进制转八进制、转十六进制（分组快速转换）。

【备考点拨】

1、掌握二进制数与十进制数的互转，熟练计算IP地址转换；

2、掌握二进制与十六进制的互转，熟练掌握内存地址转换。

#### 2.1.2 原码/反码/补码/移码表示

【考法分析】

本考点的主要查考方式有：给定一些描述，让考生判断是否正确；计算某种码制的表示范围或表示数值的个数；不同码制的表示。

【要点分析】

1、原码/反码/补码/移码的转换规则：

**原码**：符号位+数值位绝对值。

**反码**：正数的反码是原码本身，负数的反码是符号位不变数值位按位取反。

**补码：**正数的补码是原码本身，负数的补码是符号位不变数值位在反码基础上**加1**。

**移码**：正数和负数的移码都是在补码基础上符号位取反。

2、原码/反码/补码/移码的表示范围（如下图所示）：

|  |  |  |  |
| --- | --- | --- | --- |
| 码制 | 定点整数 | 定点小数 | 表示数码个数 |
| 原码 | -（2n-1-1）～+（2n-1-1） | -（1-2 -（n-1））～+（1-2 -（n-1）） | 2n-1 |
| 反码 | -（2n-1-1）～+（2n-1-1） | -（1-2 -（n-1））～+（1-2 -（n-1）） | 2n-1 |
| 补码 | - 2n-1～+（2n-1-1） | -1～+（1-2 -（n-1）） | 2n |
| 移码 | - 2n-1～+（2n-1-1） | -1～+（1-2 -（n-1）） | 2n |

【定义-0的补码为其-2n-1的补码，以n=8为例，**人为定义**10000000是-128的补码】

3、对于0的特殊化：0在原码和反码中有+0和-0两种表示方式，而对于补码和移码其表示方式一致。

【备考点拨】

1、掌握原码/反码/补码/移码的准换；

2、掌握原码/反码/补码/移码的表示范围和能够表示数的个数，也可能与浮点数表示结合考查浮点数的表示范围；

3、注意在补码和移码中-0的特殊性和人为定义。

#### 2.1.3 浮点数的表示

【考法分析】

本知识点的考查方式为：给定一些描述（关于浮点数构成、浮点数运算规则等）让学员判断是否正确；判断尾数和解码对浮点数的影响。

【要点分析】

1、浮点数各部分表示意义：N = 尾数\*基数指数

（1）一般尾数用补码，阶码用移码；

（2）阶码的位数决定数的表示范围，位数越多范围越大；

（3）尾数的位数决定数的有效精度，位数越多精度越高。

2、浮点数运算规则：对阶>尾数计算>结果格式化

（1）对阶时，小数向大数看齐；

（2）对阶是通过较小数的尾数右移实现的。

【备考点拨】

1、掌握浮点数各部分的意义，重点掌握阶码、尾数的意义；

2、掌握浮点数运算的过程和简单规则。

#### 2.1.4 逻辑运算

【考法分析】

本知识点的考查形式为：给定一些运算表达式让学员计算结果或找到等价表达形式；判断短路运算。

【要点分析】

1、运算符；

较高优先级（关系运算符）：<（小于）；<= （小于或等于）；>（大于）；>= （大于或等于）。

较低优先级（关系运算符）：==等于；!=不等于。

逻辑运算符：

逻辑或（ ||、+ 、∪、∨、OR）：连接的两个逻辑值全0时才取0。

逻辑与（&&、\*、・、∩、∧、AND ）：连接的两个逻辑值全1时才取1。

逻辑异或（⊕、XOR ）：连接的两个逻辑值不相同时才取1，相同则取0。

逻辑非（！、﹃ 、～ 、NOT，—）：将原逻辑值取反即可。

2、优先级

！（非）->&&（与）->||（或）

逻辑运算符中的“&&”和 “||”低于关系运算符，“!”高于算术运算符

因此运算符的优先顺序为：！>算术运算符>关系运算符>&&>||>赋值运算符

3、短路原则：在逻辑表达式的求解中，并不是所有的逻辑运算符都要被执行。

（1）a&&b&&c 只有a为真时，才需要判断b的值，只有a和b都为真时，才需要判断c的值。（2）a||b||c 只要a为真，就不必判断b和c的值，只有a为假，才判断b。a和b都为假才判断c。

【备考点拨】

1、掌握运算符写法和计算规则；

2、掌握短路原则。

### 2.2 运算器与控制器（★★★★）

【考法分析】

本考点主要的考查形式有：给定部件判断部件所属运算器或控制器；给定一定功能描述，判断具体描述的部件。

【要点分析】

1、运算器：

①**算术逻辑单元ALU**：数据的算术运算和逻辑运算；

②**累加寄存器AC**：通用寄存器，为ALU提供一个工作区，用在**暂存**数据；

③**数据缓冲寄存器DR**：写内存时，暂存**指令或数据**；

④**状态条件寄存器PSW**：存状态标志与控制标志（争议：也有将其归为控制器的）。

2、控制器：

①**程序计数器PC**：存储**下一条**要执行指令的地址；(先执行)

②**指令寄存器IR**：存储**即将**执行的指令；(后执行)

③**指令译码器ID**：对指令中的操作码字段进行分析解释；

④时序部件：提供时序控制信号。

【备考点拨】

1、掌握运算器、控制器分类所属的部件，并熟悉各个部件的功能和特点能够加以区分。

### 2.3 Flynn分类法（★★）

【考法分析】

本题考查形式主要是根据题干描述判断所属分类或判断各个分类的典型系统。

【要点分析】

|  |  |  |  |
| --- | --- | --- | --- |
| 体系结构类型 | 结构 | 关键特性 | 代表 |
| 单指令流单数据流  SISD  (Single Instruction Single Data stream) | 控制部分：一个  处 理 器：一个  主存模块：一个 |  | 单处理器系统 |
| 单指令流多数据流  SIMD  (Multiple ) | 控制部分：一个  处 理 器：多个  主存模块：多个 | 各处理器以异步的形式执行同一条指令 | 并行处理机  阵列处理机  超级向量处理机 |
| 多指令流单数据流  MISD | 控制部分：多个  处 理 器：一个  主存模块：多个 | 被证明不可能，至少是不实际 | 目前没有，有文献称流水线计算机为此类 |
| 多指令流多数据流  MIMD | 控制部分：多个  处 理 器：多个  主存模块：多个 | 能够实现作业、任务、指令等各级全面并行 | 多处理机系统  多计算机 |

【备考点拨】

1、掌握Flynn分类法四种类型的特点及其相关典型系统。

2、常考典型系统：阵列处理机、超级向量处理机；常考类型：SIMD、MISD。

### 2.4 CISC与RISC（★★）

【考法分析】

本考点的基本考法是给出一些CISC与RISC的特点，让考生判断说法的正确性；给出一定描述判断寻址方式。

【要点分析】

1、CISC：复杂指令集；RISC：精简指令集。(Complex/Reduced Instruction Set Computer)

2、CISC的特点：指令数量多，指令频率差别大，多寻址，使用微码。

3、RISC的特点：指令数量少，操作寄存器，单周期，少寻址，多通用寄存器，硬布线逻辑控制，适用于流水线。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 指令系统类型 | **指令** | 寻址方式 | 实现方式 | 其他 |
| CISC（复杂） | 数量多，使用频率差别大，可变长格式 | 支持多种 | 微程序控制技术（微码） | 研制周期长 |
| RISC（精简） | 数量少，使用频率接近，定长格式，大部分为单周期指令，操作寄存器，只有Load/Store操作内存 | 支持方式少 | 增加了通用寄存器；硬布线逻辑控制为主；适合采用流水线 | 优化编译，有效支持高级语言 |

4、寻址方式(不同寻址方式是为了扩大寻址空间并提高编程灵活性)

立即寻址方式

特点：操作数**直接**在指令中，速度快，灵活性差。

直接寻址方式

特点：指令中存放的是**操作数的地址**。

间接寻址方式

特点：指令中存放了一个地址，这个地址对应的内容是操作数的地址。

寄存器寻址方式

特点：寄存器存放操作数。

寄存器间接寻址方式

特点：寄存器内存放的是操作数的地址。

【备考点拨】

1、熟记CISC与RISC特点差异。

2、识别寻址方式。

### 2.5 流水线技术（★★★★）

【考法分析】

本考点涉及的考查形式有：

（1）流水线相关理论概念；

（2）流水线相关计算。

【要点分析】

流水线相关概念：

（1）流水线是指在程序执行时多条指令重叠进行操作的一种准并行处理实现技术。各种部件同时处理是针对不同指令而言的，它们可同时为多条指令的不同部分进行工作，以提高各部件的利用率和指令的平均执行速度。

（2）流水线建立时间：1条指令执行时间。

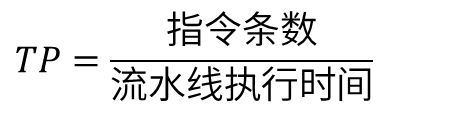
（3）流水线**周期**：执行时间**最长**的一段，记作t。

2、流水线相关计算：

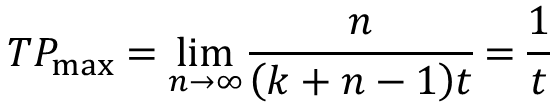
（1）流水线执行时间（理论公式）：(t1+t2+…+tk)+(n-1)\*t

（2）流水线执行时间（实践公式）：k\*t +(n-1)\*t kt一条流水线执行时间

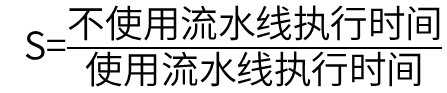
（3）流水线吞吐率：



（4）流水线最大吞吐率



（5）流水线加速比：

kt-(n-1)t

【备考点拨】

1、了解流水线相关理论概念；

2、掌握流水线相关计算公式。

### 2.6 存储系统（★★★★）

#### 2.6.1 分级存储体系

【考法分析】

本考点主要考查的形式一般为判断相关概念描述的正确性，考查较为频繁的概念有局部性原理、各层次的速度比较、Cache的位置；识别部分存储器。

【要点分析】

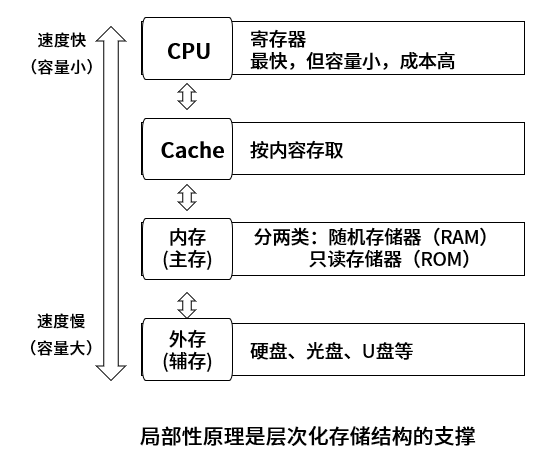
（1）关于多级存储体系的分布（见下图）；

（2）关于多级存储体系不同层次的大小、速度等对比（见下图）；

（3）关于多级存储体系的理论支持体系（局部性原理）。

**时间局部性**：指程序中的某条指令一旦执行，不久以后该指令可能再次执行，典型原因是由于程序中存在着大量的**循环操作**。（**相邻时间访问同一数据**）

**空间局部性**：指一旦程序访问了某个存储单元，不久以后，其附近的存储单元也将被访问，即程序在一段时间内所访问的地址可能集中在一定的范围内，其典型情况是程序**顺序执行**。（**相邻空间地址连续被访问**）



Cpu访问最快的是通用寄存器; 主存和辅存构成虚拟存储

（4）存储器分类

1、存储器位置

内存&外存

2、存取方式

（1）按内容存取：

相联存储器（如Cache）

（2）按地址存取：

随机存取存储器（如内存）

顺序存取存储器（如磁带）

直接存取存储器（如磁盘）

3、工作方式

（1）随机存取存储器RAM（如内存DRAM）

（2）只读存储器ROM（如BIOS）

【备考点拨】

1、掌握相关理论概念。

2、了解典型存储部件对应的存储器类型。

#### 2.6.2 Cache

【考法分析】

一般以选择题形式考查，要求判断相关概念描述是否正确，Cache映射方式的对比区分，Cache映射方式的控制由硬件直接控制。

【要点分析】

1、Cache的相关概念：理论依据（局部性原理），大小、速度、成本等对比。

在计算机的存储系统体系中，Cache是（除寄存器以外）访问速度最快的层次。

Cache改善了系统性能，提高了CPU访问内存的效率。

**Cache的地址映像由硬件直接完成**。（常考）

2、Cache映射方式：全相联、组相联、直接相联映像的对比（冲突率依次增高，电路复杂度依次降低）：

**直接相联映像**：硬件电路较简单，但冲突率很高。Cache对应页号位置已有数据即冲突需淘汰页面再调入。

**全相联映像**：电路难于设计和实现，只适用于小容量的Cache，冲突率较低。所有位置都有数据，才会产生冲突，此时才需要淘汰已有页面，然后调入新的页面

**组相联映像**：直接相联与全相联的折中。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 冲突率  （高、中、低） | 电路复杂度  （复杂、简单、折中） | 其他 |
| 直接相联映象 | 高 | 简单 | 对应位置有数据即冲突 |
| 全相联映象 | 低 | 复杂 | 所有位置有数据即冲突 |
| 组相联映象 | 中 | 折中 |  |

【备考点拨】

1、掌握Cache相关概念；

2、掌握Cache相联映象方式的对比。

#### 2.6.4 内存

【考法分析】 1byte = 1B = 1Byte = 1 字节 = 8bit = 8位

本知识点的考查主要是对相关参数的计算：内存总容量、芯片单位容量、芯片片数。总片数＝总容量/每片的容量。

【要点分析】

1、内存单元数计算：最大地址+1-最小地址

2、内存总容量：按字节编址，内存单元数\*8bit；按字编址，内存单元数\*机器字长。

3、已知芯片单位容量，求所用芯片的片数，总容量/单位容量；

4、已知所用芯片的片数，求取芯片单位容量，总容量/芯片片数。

【备考点拨】

1、掌握相关计算，注意结合二进制与十六进制、十进制之间的相互转化，注意编址方式的选择会影响内存总容量。

### 2.7 总线系统（★）

【考法分析】

本知识点考查形式主要是判断总线的分类。

【要点分析】

1、数据总线（Data Bus）：在CPU与RAM之间来回传送需要处理或是需要储存的数据。

2、地址总线（Address Bus）：用来指定在RAM（Random Access Memory）之中储存的数据的地址。

3、控制总线（Control Bus）：将微处理器控制单元（Control Unit）的信号，传送到周边设备。

【备考点拨】

1、掌握总线的分类，能够区分数据总线、控制总线、地址总线的区别。

### 2.8 可靠性R（★）

【考法分析】

本知识点主要考查关于不同系统类型的可靠性计算；识别可靠性/可用性、可维护性指标表示。

【要点分析】 可靠性R

1、串联系统计算：R总=R1\*R2\*…\*Rn;

2、并联系统计算：R总=1-(1-R1)(1-R2)…(1-Rn)； 一个并联: 1-(1-R)(1-R)

3、N模混联系统：先将整个系统划分为多个部分串联R1、R2…等，再计算R1、R2内部的并联可靠性，带入原公式。

4、可靠性指标



平均无故障时间→ (MTTF) MTTF=1/λ，λ为失效率

平均故障修复时间→ (MTTR) MTTR=1/μ，μ为修复率

平均故障间隔时间→ (MTBF) MTBF = MTTR + MTTF

系统可用性→ MTTF/(MTTR+MTTF)×100%

在实际应用中，一般MTTR很小，所以通常认为MTBF≈MTTF。

可靠性可以用可以用MTTF/(1+MTTF)来度量。

可维护性可以用1/(1+MTTR)来度量

【备考点拨】

1、掌握可靠性的计算。

2、了解可靠性/可用性、可维护性的指标表示。

### 2.9 校验码（★★★）

【考法分析】

本考点主要考查形式有：给定校验码相关基本概念，判断正误；区分不同校验方式的特点；对于海明校验码的计算问题，包括校验码的位数计算，校验码的位置计算。

【要点分析】

1、奇偶校验：掌握校验原则等相关概念，只检奇数位错，不能纠错。

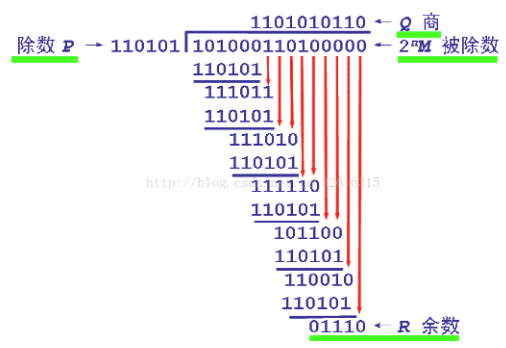
2、循环校验码CRC：可查错，不可纠错，运用模二除法计算校验码。

3、海明校验：要求掌握相关概念，可查错，可纠错；要求掌握海明校验码校验位计算：2r>=r+m+1。

4、校验码对比

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 校验码位数 | 校验码位置 | 检错 | 纠错 | 校验方式 |
| 奇偶校验 | 1 | 一般拼接在头部 | 可检奇数位错 | 不可  纠错 | 奇校验：最终1的个数是奇数个；  偶校验：最终1的个数是偶数个； |
| CRC循环冗余校验 | 生成多项式最高次幂决定 | 拼接在信息位尾部 | 可检错 | 不可  纠错 | 模二除法求余数，拼接作为校验位 |
| 海明校验 | 2r≥m+r+1 | 插入在信息位中间 | 可检错 | 可纠错 | 分组奇偶校验 |

CRC循环冗余校验:设需要发送的信息为M = 1010001101，CRC8=X5+X4+X2+1产生多项式对应的代码为P = 110101，R＝5。在M后加5个0为计算序列：101000110100000，然后对P做模2除法运算，得余数r(x)对应的代码：01110。故实际需要发送的数据是101000110101110。其中CRC校验码就为：01110。



【备考点拨】

1、掌握奇偶校验的规则，掌握其特性；

2、了解CRC校验的规则，了解模二除法，掌握其特性；

3、掌握海明校验的规则，重点掌握其校验位计算公式2r>=r+m+1。了解其编码过程。

### 2.10 计算机性能指标（★）

【考法分析】

本考点主要考查形式有：给出描述，判断对应的性能指标；给出一定参数，计算性能指标的参数。

【要点分析】

1、**主频**（计算机参数），**时钟周期=主频的倒数**，主频=倍频\*外频。

2、平均每条指令的平均时钟周期个数（**CPI**，clock per instruction），CPI=时钟周期总数/指令总条数。

3、每（时钟）周期运行指令条数（**IPC**，instruction per clock），IPC=指令总条数/时钟周期总数

4、百万条指令每秒（**MIPS**，Million Instructions Per Second）=(IPC\*时钟周期)/106

5、每秒百万个浮点操作（**MFLOPS**，Million Floating-point Operations per Second），与MIPS相似，针对的是浮点操作。

6、字长（计算机参数）。

7、**总线宽度**：每次脉冲通过的数据量。

8、**带宽**：单位时间通过的数据量，带宽=数据总量/总时间

9、**吞吐量**，某个时间段内完成的任务总数；吞吐率，单位时间内完成的任务总数，吞吐率=任务总数/总时间。

【备考点拨】

1、了解常见计算机性能指标的描述和意义；

2、了解常见参数的计算。

## 3 章节问答

1、浮点数的规格化需要掌握吗？

答：

浮点数只需要掌握运算的逻辑过程有规格化，具体的规格化过程在软设中没有涉及。

2、状态条件寄存器PSW属于运算器还是控制器？

答：

PSW在运算过程中会保存为0标识、溢出标识等，而这些标识在某些过程中会用来作为控制跳转的条件，所以在分类过程中，PSW既可以归于运算器，也可以归于控制器，存在争议。建议做题时如果碰到分类问题，将PSW放到最后考虑。

3、流水线执行时间计算有理论公式和实践公式，应该选择哪一个进行计算呢？

答：

计算流水线执行时间过程中，默认选择理论公式，在做题中，如果理论公式没有正确答案，才会考虑用实践公式计算。

4、Cache的地址映射是程序员通过应用程序控制的吗？

答：

Cache对于程序员来说是不用了解、不会使用的存储层次，也就是说Cache对于程序员来说是透明的。Cache与内存之前的映射是由硬件直接完成的。

# 第2章 操作系统

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解进程状态变迁。

（2）掌握PV操作相关解题技巧。

（3）了解死锁的条件，掌握死锁资源分配计算、银行家算法。

（4）了解页式存储、段式存储、段页式存储的特点，掌握页式存储页表的使用以及逻辑地址与物理地址的转换，掌握段式存储合法段地址的判断。

（5）掌握磁盘相关解题技巧。（磁盘存取时间计算，磁盘优化存储问题，移臂调度算法问题）

（6）掌握索引文件相关计算。

（7）掌握树形目录绝对路径与相对路径区分，了解文件名写法。

（8）掌握位示图相关计算。

（9）区分I/O控制方式（程序控制（查询）方式、程序中断、DMA）特点。

### 1.1 本章重点

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 知识领域 | | 知识点详情 |
| 1 | 进程管理 | 进程的状态（★★） | 进程的状态 |
| 2 | 前趋图（★★★） | 前趋图 |
| 3 | 信号量与PV操作（★★★★） | 信号量与PV操作 |
| 4 | 死锁及银行家算法（★★★★） | 死锁及银行家算法 |
| 1 | 存储管理 | 段页式存储（★★★★） | 页式存储 |
| 2 | 段式存储 |
| 3 | 段页式存储 |
| 4 | 页面置换算法（★） | 页面置换算法 |
| 5 | 磁盘管理（★★） | 磁盘管理 |
| 1 | 文件管理 | 绝对路径与相对路径（★★★） | 绝对路径与相对路径 |
| 1 | 索引文件（★★） | 索引文件 |
| 2 | 位示图（★★） | 位示图 |
| 1 | 设备管理 | I/O设备管理（★） | I/O设备管理 |

## 2 考点精讲

【操作系统概述】

2.1 操作系统相关概念

【考法分析】

本考点主要考查形式是识别操作系统层次、判断操作系统相关描述是否正确、判断线程部分内容是否能够共享。

【要点分析】

1、软件分层：



2、操作系统功能：

管理系统的硬件、软件、数据资源。

控制程序运行。

人机之间的接口。

应用软件与硬件之间的接口。

3、特殊的操作系统

|  |  |
| --- | --- |
| 分类 | 特点 |
| 批处理操作系统 | 单道批：一次一个作业入内存，作业由程序、数据、作业说明书组成  多道批：一次多个作业入内存，特点：多道、宏观上并行微观上串行 |
| 分时操作系统 | 采用时间片轮转的方式为多个用户提供服务，每个用户感觉独占系统  特点：多路性、独立性、交互性和及时性 |
| 实时操作系统 | 实时控制系统和实时信息系统  交互能力要求不高，可靠性要求高（规定时间内响应并处理） |
| 网络操作系统 | 方便有效共享网络资源，提供服务软件和有关协议的集合  主要的网络操作系统有：Unix、Linux和Windows Server系统 |
| 分布式操作系统 | 任意两台计算机可以通过通信交换信息  是网络操作系统的更高级形式，具有透明性、可靠性和高性能等特性 |
| 微机操作系统 | Windows：Microsoft开发的图形用户界面、多任务、多线程操作系统  Linux：免费使用和自由传播的类Unix操作系统，多用户、多任务、多线程和多CPU的操作系统 |
| 嵌入式操作系统 | 运行在智能芯片环境中  特点：微型化、可定制（针对硬件变化配置）、实时性、可靠性、易移植性（HAL和BSP支持） |

4、嵌入式操作系统特点

微型化。从性能和成本角度考虑，希望占用资源和系统代码量少，如内存少、 字长短、运行速度有限、能源少（用微小型电池）。

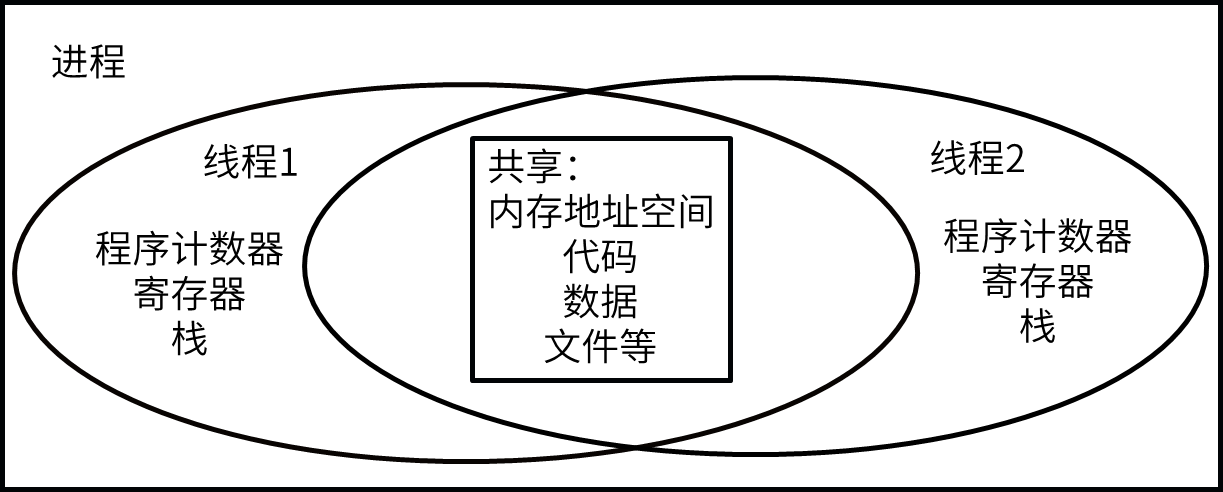
可定制。从减少成本和缩短研发周期考虑，要求嵌入式操作系统能运行在不同的微处理器平台上，能针对硬件变化进行结构与功能上的配置，以满足不同应用需要。

实时性。嵌入式操作系统主要应用于过程控制、数据采集、传输通信、多媒体信息及关键要害领域需要迅速响应的场合，所以对实时性要求高。

可靠性。系统构件、模块和体系结构必须达到应有的可靠性，对关键要害应用还要提供容错和防故障措施。

易移植性。为了提高系统的易移植性，通常采用硬件抽象层 （Hardware Abstraction Level，HAL） 和板级支持包 （Board Support Package，BSP） 的底层设计技术。

5、线程共享内容



【备考点拨】

1、了解操作系统的分层和任务。

2、了解特殊的操作系统分类和相应的一些特点。

3、了解线程能够共享的部分内容。

**【进程管理】**

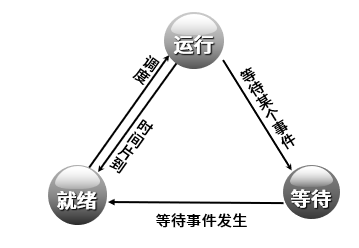
### 2.1 进程的状态（★★）

【考法分析】

本考点主要考查形式是根据图示判断相关状态位置或状态变迁条件。

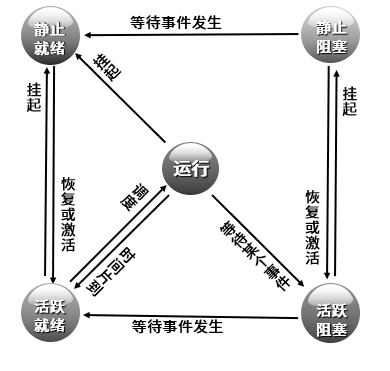
【要点分析】

操作系统三态模型如下图所示：



注：运行→等待，等待→就绪，前者等待某个事件，后者是等待的这个事件发生了，注意区分。

操作系统五态模型：



【备考点拨】

1、掌握操作系统三态模型、五态模型的状态位置及其状态变迁条件。

### 2.2 前趋图（★★★）

【考法分析】

本考点主要的考查形式有：与PV操作结合考查。

【要点分析】

1、前趋活动和后继活动：在前趋图中，前趋活动完成后通知所有后继活动；后继活动开始之前要检查是否前趋活动已经全部完成。

【备考点拨】

1、掌握前趋图相关表示的活动之间的依赖关系。

### 2.3 信号量与PV操作（★★★★）

【考法分析】

本知识点的考查形式有：单独考查信号量与PV操作控制进程的互斥和并发；结合前趋图考查某个位置对应的P、V操作及其信号量；根据题干描述的业务逻辑判断对应位置的P、V操作及其信号量。

【要点分析】

1、相关概念：互斥、同步、临界资源、临界区、信号量。

**互斥**：如千军万马过独木桥，同类资源的竞争关系。

**同步**：速度有差异，在一定情况停下等待，进程间的协作关系。

**临界资源**：诸进程间需要互斥方式对其进行共享的资源，如打印机、 磁带机等。

**临界区**：每个进程中访问临界资源的那段代码称为临界区。

**信号量**：是一种特殊的变量，表示资源数。当信号量小于0时，还可以表示排队进程数。

2、PV操作对应的过程（如下图所示）：



3、前趋图与PV操作结合，根据前趋图箭线标注信号量，再根据进程图填空：

针对箭线标注信号量，**箭线的起点位置是V操作**（即前趋活动**完成后**以V操作通知后继活动）；**箭线的终点位置是P操作**（即后继活动**开始前**以P操作检查前趋活动是否完成）。

【备考点拨】

1、掌握PV操作及信号量相关的概念；

2、理解PV操作的原理和应用，学会利用相关解题技巧，解决PV操作与信号量、前趋图的结合考查。

### 2.4 死锁及银行家算法（★★★★）

【考法分析】

本知识点主要考查形式有：根据进程情况计算死锁资源数；根据资源数利用银行家算法进行资源分配，判断选项中给出的序列是否安全。

【要点分析】

1、了解死锁的条件和预防概念：进程管理是操作系统的核心，但如果设计不当，就会出现死锁的问题。如果一个进程在等待一件不可能发生的事，则进程就死锁了。而如果一个或多个进程产生死锁，就会造成系统死锁。

2、根据题干给出的进程和资源分配，判断形成死锁的最小资源数或其他参数：对于这种情况，分配资源时每个进程得到可以完成进程的资源数减一，此时是形成死锁的最差情况，在此情况下多1个资源即可解决死锁问题，即不可能形成死锁。假设m个进程各自需要w个R资源，系统中共有n个R资源，此时不可能形成死锁的条件是：

m\*(w-1)+1<=n。

3、银行家算法：当一个进程对资源的最大需求量不超过系统中的资源数时可以接纳该进程。进程可以分期请求资源，但请求的总数不能超过最大需求量。当系统现有的资源不能满足进程尚需资源数时，对进程的请求可以推迟分配，但总能使进程在有限的时间里得到资源。根据银行家算法判断相关进程序列是否会形成死锁，是则为不安全序列。

【备考点拨】

1、掌握相关概念；

2、掌握死锁资源数计算；

3、掌握银行家算法分配资源时判断安全序列。

**【存储管理】**

### 2.5 段页式存储（★★★★）

【考法分析】

本知识点主要考查形式有：页式存储中对应逻辑页的物理页号，或对应逻辑地址的物理地址；段式存储中对应段地址的合法性判断；页式存储、段式存储、段页式存储的一些概念描述判断正误。

【要点分析】

1、知道页面大小时，可以依此判断页内地址的长度，并据此知道该地址的页号；

2、页号与页帧号的转换可以通过查表进行；

3、段地址的格式，段号后跟段内地址不能超过段长；

4、页式存储：将程序与内存均划分为同样大小的块，以页为单位将程序调入内存。

5、段式存储：按用户作业中的自然段来划分逻辑空间，然后调入内存，段的长度可以不一样。

6、段页式存储：段式与页式的综合体。先分段，再分页。1个程序有若干个段，每个段中可以有若干页，每个页的大小相同，但每个段的大小不同。

【备考点拨】

1、掌握段页式存储相关的一些概念；

2、掌握页式存储地址的转换和页表的查找；

3、掌握段式存储段地址合法性判断。

### 2.6 页面置换算法（★）

【考法分析】

本知识点主要与页式存储结合考查，依据最近最少被使用原则选择应该被淘汰的页面。

【要点分析】

1、页面淘汰时，主要依据原则（考试中默认按照此原则进行淘汰）：先淘汰最近未被访问的（访问位为0），其次淘汰被访问但未被修改的（即修改位为0，因为修改后的页面淘汰时代价更大）。

2、页面淘汰算法有多种，常用的是LRU即最近最少使用原则，依据的是局部性原理。

3、对于多种淘汰算法：最优算法OPT（理想型），随机算法RAND（随机性），先进先出FIFO（可能产生“抖动”），最近最少使用LRU（依据局部性原理）。

【备考点拨】

1、掌握页表字段表示的意义，根据LRU进行页面淘汰；

2、了解多种淘汰算法的原则，根据它们的特点进行区分。

### 2.7 磁盘管理（★★）

【考法分析】

本知识点的考查形式有：计算磁盘数据的读取时间；优化存储后的数据读取时间；磁盘调度算法的相关应用。

【要点分析】

1、存取时间=寻道时间+等待时间，寻道时间是指磁头移动到磁道所需的时间；等待时间为等待读写的扇区转到磁头下方所用的时间。有时还需要加上数据的传输时间。

2、在处理过程中，如果有关于缓冲区的使用，需要了解对于单缓冲区每次只能被一个进程使用，即向缓冲区传输数据的时候不能从缓冲区读取数据，反之亦然。

3、对于磁盘存储的优化，是因为磁头保持转动的状态，当读取数据传输或处理时，磁头会移动到超前的位置，需要继续旋转才能回到逻辑下一磁盘块，优化存储就是调整磁盘块的位置，让逻辑下一磁盘块放到磁头将要开始读取该逻辑块的位置。

4、磁盘调度算法（考虑移臂时，只需要考虑柱面信息即可，扇区不需要考虑，同一柱面的多个扇区先后顺序随机）：

先来先服务FCFS（谁先申请先服务谁）；

最短寻道时间优先SSTF（申请时判断与磁头当前位置的距离，谁短先服务谁）；

扫描算法SCAN（电梯算法，双向扫描）；

循环扫描CSCAN（单向扫描）。

【备考点拨】

1、掌握读取磁盘数据时间计算方法；

2、掌握磁盘存储优化的过程和计算方法；

3、了解磁盘调度算法的区别，并能加以区分。

**【文件管理】**

### 2.8 绝对路径与相对路径（★★★）

【考法分析】

本知识点主要考查形式即给出图示，要求选择正确的绝对路径、相对路径、文件全名。

【要点分析】

1、绝对路径从根目录开始写起，并且该文件的全名即为绝对路径+文件名。

2、相对路径从**当前位置下一级目录**开始写起。

【备考点拨】

1、掌握绝对路径和相对路径的写法。

2、掌握树形目录的一些特点和概念。

### 2.9 索引文件（★★）

【考法分析】

本知识点的考查形式主要是具体逻辑块号的索引方式判断，以及索引方式所能表示的文件大小，中间会涉及到计算。

【要点分析】

1、索引结点对应的索引方式一般题干会给出，没有给出的默认按照如图所示方式理解，下面的文件大小依图给出计算过程。

2、根据物理块大小（假设1KB）和地址项长度（假设4B），可以计算存放间接索引的物理块可以存放的地址项个数：物理块大小/地址项长度，向下取整（1KB/4B=256，注意单位和进制转换）。

3、直接索引（即索引结点直接指向实际存储文件的物理块），能够表示的逻辑页号范围是0~9，能够表示的文件大小时10\*1KB。

4、一级间接索引（即索引结点指向的物理块存放的是地址项，对应地址项个数256个，可以指向256个实际存储文件的物理块），能够表示的逻辑页号范围是10~265，能够表示的文件大小是256\*1KB。

5、二级间接索引（即索引结点指向的物理块存放的是间接索引的地址项，共256个，可以指向256个存放地址项的物理块，每个物理块指向实际存储文件的地址项有256个，最终指向的物理块共有256\*256个），能够表示的逻辑页号范围是266~65801，能够表示的文件大小是65536KB。



【备考点拨】

1、掌握索引文件的具体对应关系及相关的一些概念描述；

2、掌握索引文件逻辑页号和物理块的对应关系；

3、掌握索引文件表示文件长度的计算。

### 2.10 位示图（★★）

【考法分析】

本知识点的主要考查方式是计算指定磁盘存放的对应字的序号或位置。

【要点分析】

1、对于位示图，每一个bit位可以表示一个磁盘的占用情况，“0”表示空闲，“1”表示占用。

2、对于字的长度与具体机器字长有关，有题目指定，假设机器字长16位，则每个字可以表示16个磁盘块的占用情况。

3、指定序号为n或第n+1个磁盘，占用情况需要用m=(n+1)/16（向上取整）个字表示，字的序号为m-1。注意其中磁盘序号、字的序号、对应位号都是从0开始，计算过程中会有加1或减1处理。

【备考点拨】

1、掌握相关的概念；

2、掌握相关的计算过程。

【设备管理】

### 2.11 I/O设备管理（★）

【考法分析】

本知识的考查形式主要是给出一些描述判断对应的传输方式，或关于I/O系统的层次对应位置。

【要点分析】

1、对于I/O传输控制方式：

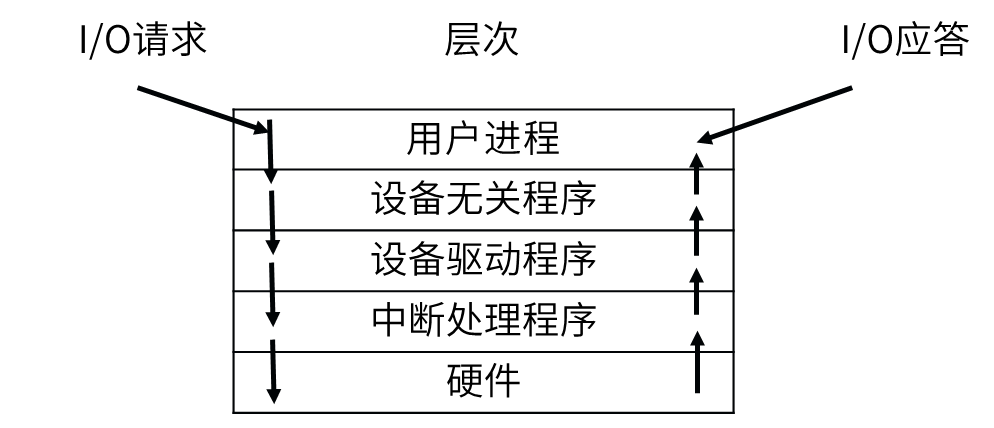
**程序查询方式**（CPU一直处于询问、等待的过程，占用CPU时间最长，CPU利用率最低）；

**中断方式**（I/O完成后向CPU发送中断请求信号，CPU和I/O可以并行）；

**DMA**（CPU只做初始化，不参与具体数据传输过程）；

通道方式、I/O处理机，专用硬件方式。

2、对于I/O软件：



硬件：完成具体的I/O操作。

中断处理程序：I/O完成后唤醒设备驱动程序

设备驱动程序：设置寄存器，检查设备状态

设备无关I/O层：设备名解析、阻塞进程、分配缓冲区

用户级I/O层：发出I/O调用。

【备考点拨】

1、掌握I/O传输控制方式的特点，能够加以区分；

2、了解I/O软件的层次和相对位置。

## 3 章节问答

1、PV操作都必须要成对出现吗？

答：

PV一定是成对出现的，PV可以简单理解为加锁和解锁，如果只加锁也就是只有P操作，可能会形成死锁。如果没有加锁，只有解锁也就是V操作，那么相当于系统中资源无端增多，也就没有意义了。

以打印机为例，如果对打印机加上PV操作，P操作相当于，每个人用之前都锁定一次打印机（S=S-1），此时其他人去打印时进行申请，会发现资源不足（S<0），此时就进入阻塞队列去排队了；而V操作相当于，能够使用打印机的人，用完之后，之前占有的打印机就会释放给其他人用（S=S+1），此时如果发现有人排队（S<=0），就会通知排队的人，可以使用了。

如果只占有不释放，很明显，排队的人只会持续暴增；如果只释放，那么信号量S无端增加1，而不会减少，也就没有意义了。

2、PV操作与互斥、同步模型有什么联系？

答：

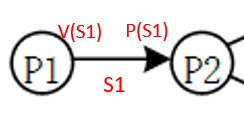
在互斥模型中，为了控制对临界资源的访问，信号量初值是临界资源的资源数，每一次使用临界资源前都会用P操作对资源进行申请锁定，用完之后立马V操作对资源进行释放，如果有进程排队，此时还会对阻塞队列进行唤醒操作。一般来说，互斥模型的PV操作，信号量针对的是临界资源，称为互斥信号量，初值为临界资源初始数量，在同一个进程中有成对的PV控制访问。

在同步模型中，为了协调多个进程，PV操作会分布在不同的进程中，此时的信号量称为同步信号量，同步模型会有等待的过程，此时的信号量初值经常会出现0的情况，也就是初始资源不足，等待其他进程制造资源并唤醒。

3、PV操作与前趋图结合考查时，为什么存在信号量比进程数还多的情况？

答：

PV操作与前趋图结合考查时，前趋图一般是同步模型，为了协调控制，此时信号量针对的不是进程，而是进程的变迁，也就是说，信号量的数量与箭线的数量是相等的，PV操作控制的是进程变迁过程。



4、移臂调度过程中，同一柱面有多个扇区怎么办？

答：

一般而言，涉及到移臂调度算法时，只需要考虑移臂过程，查看柱面顺序即可。旋转等待时间跟多磁头（盘面）是否并发读写有关系，从目前磁盘技术来看，是支持多个盘面同时读写的。 所以只要是同一柱面，扇区顺序没有要求。

5、磁盘读取数据过程中单缓冲区和双缓冲区有什么区别？

答：磁盘读取涉及到缓冲区一般形式如下：



处理数据可以理解为三个处理步骤：

①从磁盘读入到缓冲区；

②从缓冲区读入到（内存）用户区；

③处理（内存）用户区数据。

如果是单缓冲区，则意味步骤①和②都需要访问临界资源——缓冲区，所以需要合并成一个操作阶段。（缓冲区同一时刻只允许一个进程访问）

如果是双缓冲区，可以实现读入到缓冲区2和从缓冲区1读入到用户区的并发。

可以构造流水线进行计算。

单缓冲区：使用缓冲区时不能并行，必须分开执行，时间为①和②的时间之和；然后处理数据。构造成流水线后，整个过程划分为2个阶段，分别是①和②的时间之和，③的时间，根据流水线执行公式进行计算。

双缓冲区：读入缓冲区，和从缓冲区读入用户区，可以对不同的缓冲区进行，也就是说，可以并行处理。对于这里构造成流水线后，整个过程划分为3个阶段，即①、②、③。根据流水线执行公式进行计算结课。

PS：注意理解这里流水线的构造过程。

# 第3章 数据库系统

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解数据库模式、数据库设计过程。

（2）熟悉E-R图图示，掌握E-R图转关系模式的原则。

（3）掌握关系代数（并、交、差、笛卡尔积、投影、选择、自然连接）。

（4）熟悉函数依赖、部分函数依赖、传递函数依赖等概念，熟悉候选码\候选键、主码\主键、外键的概念，掌握候选码的求取，区分主属性和非主属性概念。

（5）了解规范化相关的问题描述和作用，掌握1NF、2NF、3NF、BCNF的判断区分。

（6）掌握模式分解对于无损、保持函数依赖的判断。

（7）了解常用的SQL语言语法，比如select用法。

（8）了解事务的特性，事务并发可能产生的问题，封锁协议（S锁和X锁）。

（9）了解数据完整性分类（数据完整性、参照完整性、自定义完整性等）。

（10）了解分布式数据库、数据库备份相关扩展知识。

（11）掌握软件设计——数据库设计部分（下午题试题二）解题技巧。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 数据库模式（★★） | 数据库模式 |
| 1 | 数据库设计过程（★★） | 数据库设计过程 |
| 1 | ER模型（★★★★★） | E-R图图元 |
| 2 | E-R图转关系模式原则 |
| 1 | 关系代数（★★★） | 关系代数（并交叉、笛卡尔积、投影、选择、自然连接） |
| 1 | 规范化理论（★★★★★） | 规范化理论 |
| 1 | SQL语言（★★★★） | SQL语言 |
| 1 | 并发控制（★★） | 并发控制 |
| 1 | 数据库完整性约束（★） | 数据库完整性约束（★） |
| 1 | 数据库扩展知识（★） | 分布式数据库 |
| 2 | 数据库备份 |
| 1 | 【软件设计】  数据库设计题解题技巧 | 数据库设计题解题技巧 |

## 2 考点精讲

### 2.1 数据库模式（★★）

【考法分析】

本知识点的主要考查方式是判断模式（外模式、模式、内模式）与产物（视图、库表、文件）的对应关系，或给定一些概念描述判断正误。

【要点分析】

1、三级模式：外模式对应视图，模式（也称为概念模式）对应数据库表，内模式对应物理文件。

2、两层映像：外模式-模式映像，模式-内模式映像；两层映像可以保证数据库中的数据具有较高的逻辑独立性和物理独立性。

3、逻辑独立性：即逻辑结构发生改变时，用户程序对外模式的调用可以不作修改；物理独立性：即数据库的内模式发生改变时，数据的逻辑结构不变。

【备考点拨】

掌握三级模式-两层映像结构和相关概念。

### 2.2 数据库设计过程（★★）

【考法分析】

本知识点主要考查形式有：在上午题中判断各个阶段的任务和产物；在下午题中数据库设计题考查思路与数据库设计过程基本一致。

【要点分析】



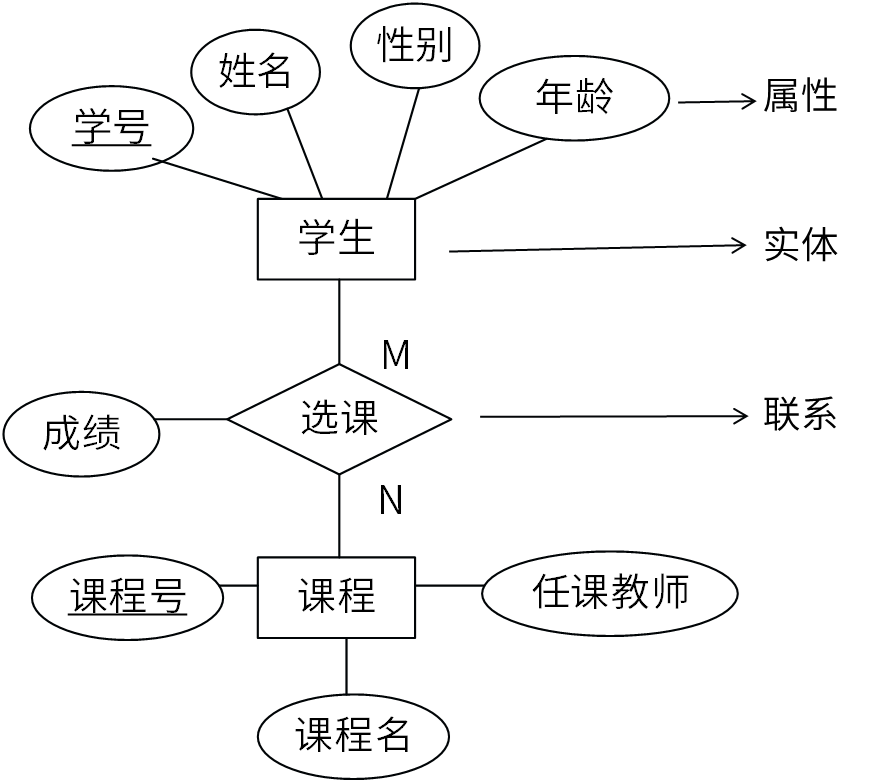
### 2.3 ER模型（★★★★★）

【考法分析】

本知识点主要考查形式有：在上午题中给出E-R图让考生判断某些部分的缺失、定义，或关系的类型判断，E-R图向关系模式的转换；在下午题中数据库设计题会考查补充E-R图，并且会涉及到E-R图向关系模式转换。

【要点分析】

#### 2.3.1 E-R图图元



1、**实体**：在E-R模型中，实体用矩形表示，通常矩形框内写明实体名。实体是现实世界中可以区别于其他对象的“事件”或“物体”。实体集是具有相同属性的实体集合。我们常见的实体意义应该是实体集。（比如某一个学生是实体，所有学生都具有一些相同的属性，因此学生的集合可以定义为一个实体集。也可以对应到具体数据库，实体集是表，而实体是其中特定的条目元组）

2、**属性**：属性是实体某方面的特性。在E-R图中用椭圆表示，并以无向边与对应实体进行连接。每个属性都有其取值范围。属性具有以下分类：

**简单属性和复合属性**。单独属性是原子的、不可再分的，复合属性可以分解为更小的部分。一般对于复合属性会在题干中给出其分解特性，比如对于地址，可进一步划分为省、市、区，若不特别声明，通常指的是简单属性。

**单值属性和多值属性**。对于一个特定的实体都只有一个特定的值，则为单值属性；在某些特殊的情况下，一些属性可能对应一组值，这样的属性为多值属性。比如对于职工家属，可能有多名成员，会有一组取值，这里即为多值属性。

**派生属性**。派生属性可以从其他属性得来。比如出生年月可以计算得出年龄，当出生年月属性已记录时，年龄可以计算出来，为派生属性。

**NULL属性**。当实体在某个属性上没有特定值或属性值未知时，使用NULL值，表示无异议或不知道。

3、**联系**

在E-R模型中，联系用菱形表示，通常菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标注上联系的类型（1:1、1:n、n:m）。

不同实体集间的二元关系：



同一实体集内部的二元联系：



三元联系：

以三元关系中的一个实体作为中心，假设另两个实体都只有一个实例：

若中心实体只有一个实例能与另两个实体的一个实例进行关联，则中心实体的连通数为“一”。

若中心实体有多于一个实例能与另两个实体实例进行关联，则中心实体的连通数为“多”。

语义表示：

P\_D：表示一个病房有多个病人和多个医生，一个医生只负责一个病房，一个病人只属于一个病房。

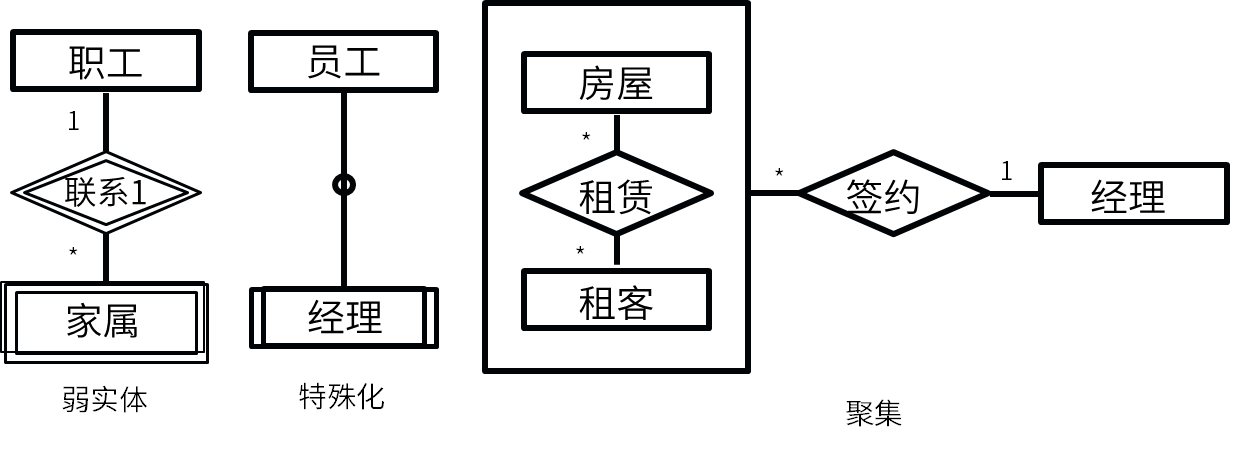
SP\_P：表示供应商为多个项目供应多种零件，每个项目可用多个供应商供应的零件，每种零件可由不同的供应商供应。

4、扩展的E-R模型

**弱实体**：在现实世界中有一种联系比较特殊，这种联系代表实体间的所有（Ownership）关系。这种实体对于另一些实体存在很强的依赖关系，即一个实体的存在必须以另一个实体为前提，将这类实体称为弱实体。比如附件是邮件的弱实体。弱实体一般以双矩形表示，与其依赖的实体之间存在联系。

**特殊化：**特殊化主要描述的是子类和父类之间的关系。子类是超类的特殊化，子类继承超类的所有属性和联系，并且还有自己特殊的属性和联系。比如经理是员工的特殊化。特殊化实体一般以双边矩形表示，与其超类以线圈的方式连接。

**聚集**：一个联系作为另一个联系的一端。



#### 2.3.2 E-R图转关系模式原则

1、每个实体型转换为一个关系模式

2、联系转关系模式：

（1）一对一联系的转换有两种方式。

**独立的关系模式**：并入两端主键及联系自身属性。（主键：任一端主键）

**归并（任意一端）**：并入另一端主键及联系自身属性。（主键：保持不变）

（2）一对多联系的转换有两种方式。

**独立的关系模式**：并入两端主键及联系自身属性。（主键：多端主键）

**归并（多端）**：并入另一端主键及联系自身属性。（主键：保持不变）

（3）多对多联系的转换只有一种方式

**独立的关系模式**：并入两端主键及联系自身属性。（主键：两端主键的组合键）

【备考点拨】

1、掌握E-R图的绘制；

2、能够正确识别实体、弱实体、属性、联系、联系类型；

3、掌握E-R图向关系模式的转换。

### 2.4 关系代数（★★★）

【考法分析】

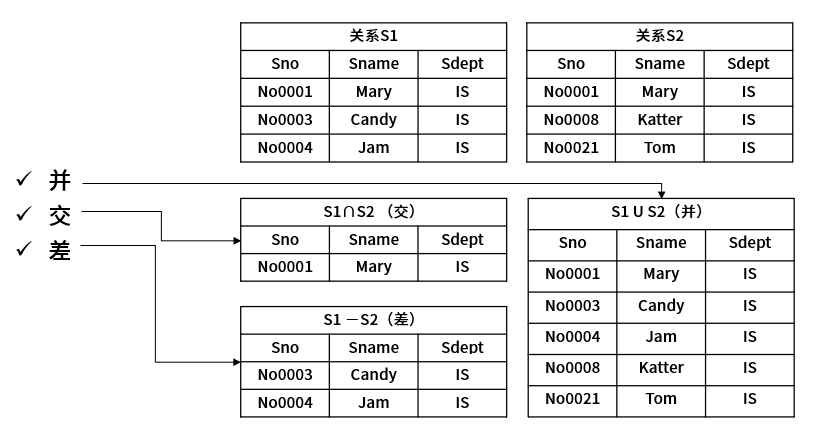
本知识点主要考查形式是：给定代数式，求取计算结果或其结果的特性，找到等价表达式，常考的关系代数是笛卡尔积、选择、投影组合与自然连接的等价表示，以及与SQL语句SELECT结合考查。

【要点分析】

1、并（结果为二者元组之和去除重复行）

2、交（结果为二者重复行）

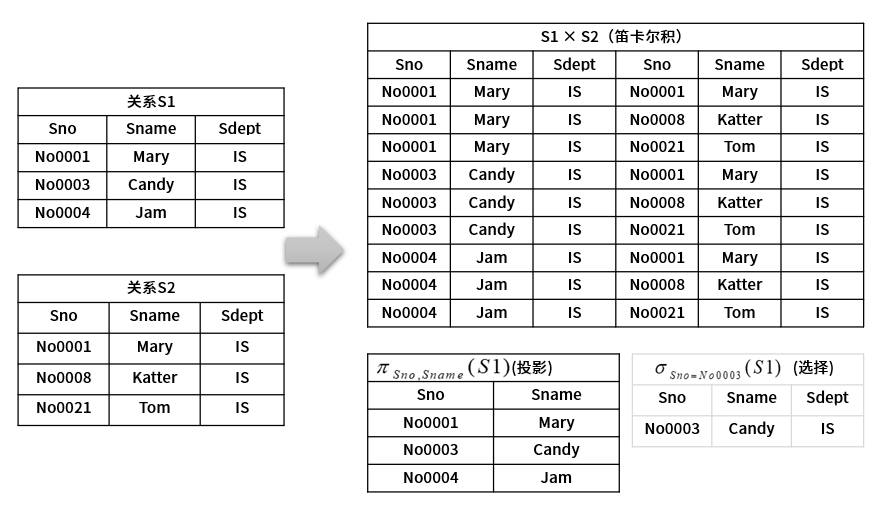
3、差（前者去除二者重复行）

类似于集合运算，计算如下图所示： 

4、笛卡尔积：结果列数为二者属性列数之和，行数为二者元素数乘积。

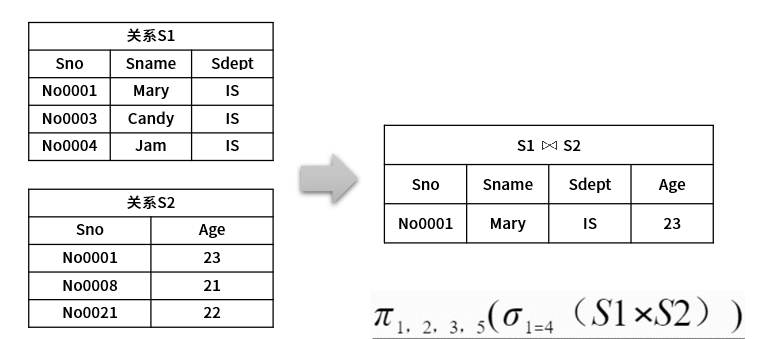
5、投影：对属性列的选择列出。

6、选择：对元组行的选择列出。

属性名可以依次标序号，直接以数字形式出现在表达式中。计算如下图所示： 

7、自然连接：结果列数为二者属性列数之和减去重复列，行数为二者同名属性列其值相同的结果元组。笛卡尔积、选择、投影的组合表示可以与自然连接等价。

普通连接的条件会写出，没有写出则表示为自然连接。计算如图所示：



【备考点拨】

1、掌握关系代数的计算。

2、关系代数经常与SELECT语句结合考查。对于常规SELECT语句格式：SELECT 语句1 FROM 语句2 WHERE 语句3，其中语句1是投影列的列名，语句2是查询的表格对象（如果有多个表格，用逗号隔开，则结果用笛卡尔积处理），语句3为选择行的条件。

### 2.5 规范化理论（★★★★★）

【考法分析】

本知识点主要的考查形式是找出某些关系中的主键、外键，判断某些关系模式的规范化程度，有时会在下午题中出现相关的提问，涉及到概念的考查。

【要点分析】

1、**候选键（候选码）**是能够唯一标示元组却无冗余的属性组合，可以有多种不同的候选键，在其中任选一个作为**主键**。候选键的求取可以利用图示法找入度为0的属性集合，并在此基础上进行扩展，最终找到能够遍历全图的最小属性组合作为候选键，对于入度为0在关系依赖集中可以理解为从未在箭线右侧出现。

2、组成候选码的属性就是**主属性**，其他为**非主属性**。

3、**外键**是其他关系模式的主键。

4、**范式**：规范化过程是为了解决数据冗余、删除异常、插入异常、更新异常等问题。

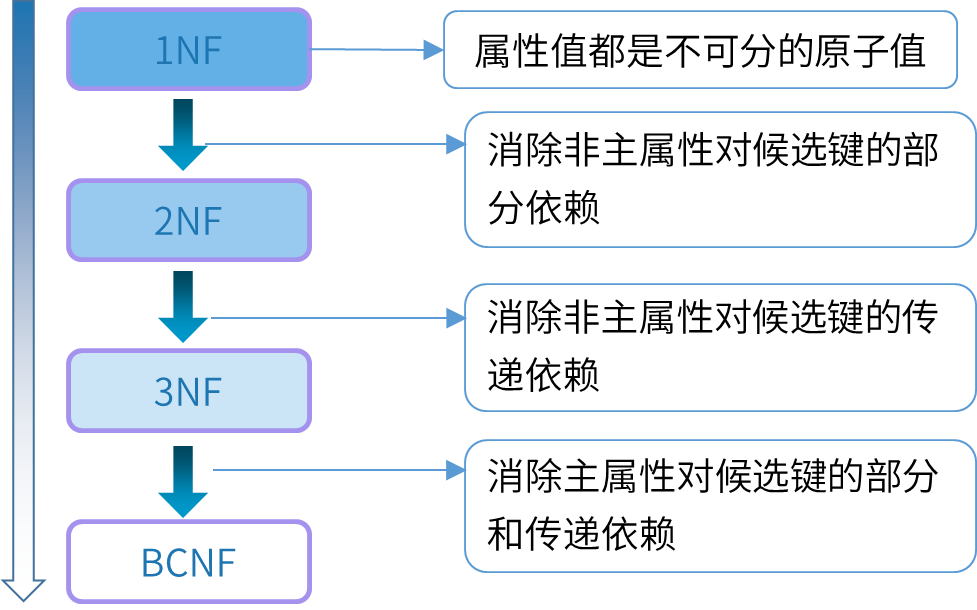
（1）**第一范式（1NF）**：在关系模式R中，当且仅当所有域只包含原子值，即每个属性都是不可再分的数据项，则称关系模式R是第一范式。

（2）**第二范式（2NF）**：当且仅当关系模式R是第一范式（1NF），且每一个非主属性完全依赖候选键（没有不完全依赖）时，则称关系模式R是第二范式。

（3）**第三范式（3NF）**：当且仅当关系模式R是第二范式（2NF），且R中没有非主属性传递依赖于候选键时，则称关系模式R是第三范式。

（4）**BC范式（BCNF）**：设R是一个关系模式，F是它的依赖集，R属于BCNF当且仅当其F中每个依赖的决定因素必定包含R的某个候选码。

规范化过程如下图所示：



5、规范化过程：分解关系模式。

（1）**保持函数依赖**：设数据库模式ρ={R1，R2，…，Rk}是关系模式R的一个分解，F是R上的函数依赖集，ρ中每个模式Ri上的FD集是Fi。如果{F1，F2，…，Fk}与F是等价的（即相互逻辑蕴涵），那么称分解ρ保持FD。

（2）**无损联接分解**：指将一个关系模式分解成若干个关系模式后，通过自然联接和投影等运算仍能还原到原来的关系模式。（表格法，公式法——仅限分解为2个子关系）

【备考点拨】

1、掌握候选键、主键、外键的求取；

2、掌握规范化理论相关的概念和规范化过程；

3、掌握关系分解后，对无损分解、保持函数依赖的判断。

### 2.6 SQL语言（★★★★）

【考法分析】

本知识点的考查形式主要有：与关系代数结合考查相关SQL语言的写法，或单纯考查SQL语言的应用。

【要点分析】

1、SQL语言

|  |  |
| --- | --- |
| 分类 | 动词 |
| 数据查询 | SELECT |
| 数据定义 | CREATE、DROP、ALTER |
| 数据操纵 | INSERT、UPDATE、DELETE |
| 数据控制 | GRANT、REVORK |

2、简单查询语句

SELECT [ALL|DISTINCT] <目标表达式> [， <目标表达式>]…

FROM <表名> [，<表名>]…

[WHERE <条件表达式>]

[ORDER BY <列名2> [ASC|DESC ] … ]；

3、分组查询

[GROUP BY <列名1> [HAVING <条件表达式> ] ]

聚集函数

|  |  |  |
| --- | --- | --- |
| 处理类型 | 处理子类 | 示例/语法 |
| 结果排序 | 升序或降序 | ORDER BY　字段名　DESC|ASC |
| 集函数 | 统计 | COUNT([DISTINCT|ALL] <列名>) |
| 计算一列中值的总和 | SUM([DISTINCT|ALL] <列名>) |
| 计算一列值的平均值 | AVG([DISTINCT|ALL] <列名>) |
| 求一列值中的最大值 | MAX([DISTINCT|ALL] <列名>) |
| 求一列值中的最小值 | MIN([DISTINCT|ALL] <列名>) |
| 对结果分组 | 将查询结果按列值分组 | GROUP BY <列名> |
| 对分组结果筛选 | 对分组结果筛选 | HAVING <条件列达式> |

4、权限控制语句

授权语句 GRANT <权限>[ ,… n]

ON <对象类型><对象名>

TO <用户>[ , … n]

WITH GRANT OPTION

WITH GRANT OPTION 子句，获得权限的用户还可以将权限赋给其他用户

收回权限语句 REVOKE<权限>[ , … n]

ON <对象类型><对象名>

FROM <用户>[ , … n]

[RESTRICT|CASCADE]

【备考点拨】

1、掌握常用的SQL语言的语法。

### 2.7 并发控制（★★）

【考法分析】

本知识点的考查形式主要是给出一些情景判断出现的并发问题，或给出一些关于事务、锁等概念的描述，判断正误。

【要点分析】

1、事务特性（ACID）：

**原子性**：事务是原子的，要么都做，要么都不做。

**一致性**：事务执行的结果必须保证数据库从一个一致性状态变到另一个一致性的状态。因此，当数据库只包含成功事务提交的结果时，称数据库处于一致性状态。

**隔离性**：事务相互隔离，当多个事务并发执行时，任一事务的更新操作直到其成功提交的整个过程，对其他事务都是不可见的。

**持续性**：一旦事务成功提交，即使数据库崩溃，其对数据库的更新操作也将永久有效。

2、并发产生的问题：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| （1）丢失更新/丢失修改 | |  | （2）不可重复读 | |  | （3）读“脏”数据 | |
| T1 | T2 |  | T1 | T2 |  | T1 | T2 |
| ①读A=10  ②  ③A=A-5写回  ④ | 读A=10  A=A-8写回 |  | ①读A=20  读B=30  求和=50  ②  ③读A=70  读B=30  求和=100  （验算不对） | 读A=20  A←A+50  写A=70 |  | ①读A=20  A←A+50  写回70  ②  ③ROLLBACK  A恢复为20 | 读A=70 |
|  |  |  |  |

3、封锁协议：

**共享锁（S锁）**：若事务T对数据对象A添加了S锁，则只允许T读取A，但不能修改A。并且其他事务只能对A加S锁，不能加X锁。

**排他锁（X锁）**：若事务T对数据对象A添加了X锁，则只允许T读取和修改A，其他事务不能再对A加任何锁。

【备考点拨】

1、掌握事务特性的概念。

2、能够区分并发产生的问题。

3、了解封锁协议的加锁原则和两种锁的特性。

### 2.8 数据库完整性约束（★）

【考法分析】

本知识点的考查形式主要是给出一定描述，判断其正误。

【要点分析】

**实体完整性**：规定其主属性不能去空值。

**参照完整性**（也称为引用完整性）：规定其外键为参照表的主键值或为空值。

**用户自定义完整性**：指用户针对某一具体的关系数据库的约束条件，反映某一具体应用所涉及的数据必须满足的予以要求，由应用的环境决定，如年龄定义为0~150正整数。

**触发器**：一种复杂的完整性约束。

【备考点拨】

1、掌握完整性约束相关的概念，能够区分不同的完整性类别，判断一些描述的正误。

### 2.9 数据库扩展知识（★）

【考法分析】

本知识点的考查形式主要是给出一定描述，判断其对应概念。

【要点分析】

**1、数据库备份分类：**从备份量来分，可以分为完全备份、增量备份、差异备份。

**完全备份**：备份所有数据。即使两个备份时间点之间数据没有任何变动，所有数据还是会被备份下来。

**增量备份**：跟完全备份不同，增量备份在做数据备份前会先判断数据的最后修改时间是否比上次备份的时间晚。如果不是，则表示该数据并没有被修改过，这次不需要备份。所以该备份方式，只记录上次备份之后的变动情况，而非完全备份。

**差异备份**：差异备份与增量备份一样，都只备份变动过的数据。但前者的备份是针对上次完整备份后，曾被更新过的。

2、分布式数据库透明性

**分片透明**：是指用户不必关心数据是如何分片的，它们对数据的操作在全局关系上进行，即如何分片对用户是透明的，因此，当分片改变时应用程序可以不变。分片透明性是最高层次的透明性，如果用户能在全局关系一级操作，则数据如何分布，如何存储等细节自不必关心，其应用程序的编写与集中式数据库相同。

**复制透明**：用户不用关心数据库在网络中各个节点的复制情况，被复制的数据的更新都由系统自动完成。在分布式数据库系统中，可以把一个场地的数据复制到其他场地存放，应用程序可以使用复制到本地的数据在本地完成分布式操作，避免通过网络传输数据，提高了系统的运行和查询效率。但是对于复制数据的更新操作，就要涉及到对所有复制数据的更新。

**位置透明**：是指用户不必知道所操作的数据放在何处，即数据分配到哪个或哪些站点存储对用户是透明的。

**局部映像透明性（逻辑透明）**：是最低层次的透明性，该透明性提供数据到局部数据库的映像，即用户不必关心局部DBMS支持哪种数据模型、使用哪种数据操纵语言，数据模型和操纵语言的转换是由系统完成的。因此，局部映像透明性对异构型和同构异质的分布式数据库系统是非常重要的。

3、分布式数据库特点

**数据独立性**。除了数据的逻辑独立性与物理独立性外，还有数据分布独立性（分布透明性）。

**集中与自治共享结合的控制结构**。各局部的DBMS可以独立地管理局部数据库，具有自治的功能。同时，系统又设有集中控制机制，协调各局部DBMS的工作，执行全局应用。

适当增加数据冗余度。在不同的场地存储同一数据的多个副本，可以提高系统的可靠性和可用性，同时也能提高系统性能。

（提高系统的**可用性**，即当系统中某个节点发生故障时，因为数据有其他副本在非故障场地上，对其他所有场地来说，数据仍然是可用的，从而保证数据的完备性。）

全局的一致性、可串行性和可恢复性。

【备考点拨】

1、了解备份的概念以及不同备份方式的区别；

2、了解分布式数据库的特点和透明性的描述。

### 2.10 【软件设计】数据库设计题解题技巧

1、补充E-R图

**找实体**：实体一般为名词形式，实体是现实世界中可以区别于其他对象的“事件”或“物体”。在E-R图中可根据相关实体间的联系，补充缺失的实体。实体以矩形表示，在矩形框内写明实体名。

**找联系**：在E-R模型中，联系用菱形表示，通常菱形框内写明联系名，并用无向边分别与有关实体连接起来，同时在无向边旁标注上联系的类型（1:1、1:n、n:m）。根据题干描述找出对应实体间存在的联系。

**三元联系**：对于题干描述中，一个联系涉及到3个实体的情况，我们称之为三元联系。以菱形表示联系，以无向边分别连接对应的实体。

2、补充关系模式

（1）根据题干描述，找遗漏属性；

（2）题干描述已完善，根据联系归并找参照属性，通常将联系归并到实体端时，需要补充另一端实体的主键，如果联系本身存在属性，归并后也需要列出。

2、判断主键和外键

（1）找主键

（2）找外键

（3）全码：关系模型的所有属性组是这个关系模式的候选码，称为全码。

3、扩展题型

**增加实体**：根据题干描述添加实体，注意新实体与其他实体之间的联系。

**完整性约束相关**：了解主键、外键相关的概念，根据题干，作出相关判断。

**规范化理论相关**：了解规范化理论相关的概念，对于规范化程度没有达到3NF时，一般认为会存在数据冗余、修改异常、插入异常、删除异常问题。对于相关问题的解决，一般是将表进行模式分解，从而提高其规范化程度至3NF。

（在软件设计师考试中，很少考查具体的BCNF规范化。）

## 3 章节问答

1、聚簇索引的建立是哪个阶段的任务？

答：

聚簇索引会将具体的物理存储改为顺序方式，所以属于内模式，物理设计阶段的任务。

2、数据模型的要素有哪些？

答：

数据模型的三要素：数据结构、数据操作和数据的约束条件。

3、联系类型的判断中，如果题干与日常生活经验冲突怎么处理？

答：

考试中，无论是上午题还是下午题，做题时都是以题干为主，当题干描述没有涉及时，才会依据日常惯例。如果有冲突，以题干为准。

4、SQL语句是否需要深入学习？

答：

从考试的角度来看，SQL在软设中的考查以SELECT语句为主，其他部分了解前面列举的一些函数即可。从实践的角度来看，如果从事相关工作，可以深入学习SQL语言。如果有时间和精力，可以自己去拓展了解。

5、关系模式如何判断其规范化程度？

答：

根据范式的判断依据，首先了解清楚一些基本概念：候选键、主属性、非主属性、部分函数依赖、传递函数依赖、函数依赖集合、函数依赖的决定因素。

然后再根据定义一步一步地深入判断：

属性不可再分则满足1NF；

在1NF基础上，如果存在**非主属性**对**候选键**的**部分函数依赖**则当前规范化程度最高只能达到1NF，如果已经消除了非主属性对候选键的部分函数依赖（**候选键只有单属性**则必定不存在对候选键的部分函数依赖），则当前规范化程度至少满足2NF；

在2NF基础上，如果存在**非主属性**对**候选键**的**传递函数依赖**则当前规范化程度最高只能达到2NF，如果已经消除了非主属性对候选键的传递函数依赖，则当前规范化程度至少满足3NF（如果**没有非主属性**，则至少满足3NF）；

在3NF基础上，如果函数依赖集合中的**所有函数依赖**，都满足其**左侧决定因素包含候选键**，则当前规范化程度至少满足BCNF，否则最高只能达到3NF。

6、模式分解是否保持函数依赖与有损/无损分解的判断，是否具有相关性？

答：

这是两个维度的独立判断二者没有相关性，不要混淆判断。

例1：对于原关系模式R（U,F），R=(A,B,C,D)，F={A→B，C→D}，分解为R1(AB)，R2(CD)，此时保持了函数依赖但是有损。

例2：对于原关系模式R（U,F），R=(A,B,C,D)，F={A→B，C→D，A→D}，分解为R3(ABC)，R2(CD)，此时没有保持函数依赖但无损。

7、模式分解如何利用表格法判断有损/无损？

答：

表格法过程：

（1）绘制初始表：原关系模式所有属性名作为列名，分解的关系模式作为第一列的值，根据分解后的关系模式，对已出现的属性记作√，未出现的属性记作×。

（2）尝试还原属性。首先查找多个关系模式中相交的属性名，接着在原函数依赖集合F中查看是否有以此属性为左侧决定因素的函数依赖存在，且该函数依赖在某个分解中已保留可用，根据这3个前提同时存在，则可以在所有存在该属性的关系模式中，将该函数依赖右侧被决定的属性还原，记作√。

（3）根据当前情况，重新尝试还原属性，重复步骤2。

（4）直到存在一个关系模式所有属性被还原，则该关系模式无损，否则有损。

以前面的例1、例2进行说明。

例1：原关系模式R（U,F），R=(A,B,C,D)，F={A→B，C→D}，分解为R1(AB)，R2(CD)

初始表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(AB) | √ | √ | × | × |
| R2(CD) | × | × | √ | √ |

尝试还原关系模式，不存在相交的属性，无法还原，因此该模式分解有损。

例2：对于原关系模式R（U,F），R=(A,B,C,D)，F={A→B，C→D，A→D}，分解为R3(ABC)，R2(CD)。

初始表如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(ABC) | √ | √ | √ | × |
| R2(CD) | × | × | √ | √ |

尝试还原关系模式：

找到相交属性名C，查看原函数依赖集合F存在C→D，且在分解R2(CD)中保留可用，3个前提同时满足，此时可以在R1中还原D属性，得到新的表格如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| R1(ABC) | √ | √ | √ | √ |
| R2(CD) | × | × | √ | √ |

R1所有属性被还原，因此该模式分解无损。

# 第4章 计算机网络与信息安全

## 1 考情分析

【计算机网络】

（1）掌握OSI/RM七层模型，了解各层涉及的硬件设备和传输单位。

（2）掌握TCP/IP四层模型，了解与七层模型对应关系，熟悉常用的协议。

（3）了解IP分类，掌握子网划分和路由汇聚计算，了解IPv6的概念。

（4）了解网络规划与设计的一些原则，了解分层结构（接入层、汇聚层、核心层）。

（5）了解计算机网络分类。

（6）了解网络接入技术。

（7）了解HTML一些常用的标签，了解URL的结构和域名的分类。

【信息安全】

（1）理解对称加密过程，了解对称加密的特点和常见的对称加密算法。

（2）理解非对称加密过程，了解非对称加密的特点和常见的非对称加密算法。

（3）理解数字签名和消息摘要的结合使用，了解摘要算法的特点和常见的摘要算法。

（4）了解PKI公钥体系的使用，了解数字证书的内容和主要部分的验证作用。

（5）了解网络安全协议分层（PGP,HTTPS,SSL,SET等）。

（6）了解网络攻击的分类，区分常见的攻击分类。

（7）了解防火墙的特点。

（8）了解安全防范体系的层次划分。

（9）了解病毒和木马分类，了解常见的一些病毒和木马名称。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | OSI七层模型（★★） | OSI七层模型 |
| 1 | TCP/IP协议簇（★★★★） | TCP/IP协议簇 |
| 1 | IP地址与子网划分（★★★★） | Flynn分类法 |
| 1 | 网络规划与设计（★） | 网络规划与设计 |
| 1 | 网络接入技术（★★） | 网络接入技术 |
| 1 | HTML（★★） | 分级存储体系 |
| 1 | URL（★） | URL |
| 1 | 对称加密与非对称加密（★★★） | 对称加密与非对称加密 |
| 1 | 信息摘要与数字签名（★★） | 信息摘要与数字签名 |
| 1 | 数字证书（★★） | 数字证书 |
| 1 | 网络安全协议（★★★） | 网络安全协议 |
| 1 | 防火墙技术与网络攻击（★★★） | 防火墙技术与网络攻击 |
| 1 | 计算机病毒与木马（★） | 计算机病毒与木马 |

## 2 考点精讲

### 2.1 开放系统互连参考模型OSI（★★）

【考法分析】

本知识点的主要考查形式有：给定物理设备判定所属层次；或给定传输单位，判断其所属层次；或给定相关功能描述，判断其所属层次。

【要点分析】

|  |  |  |  |
| --- | --- | --- | --- |
| 层次 | 名称 | 主要功能 | 主要设备及协议 |
| 7 | 应用层 | 实现具体的应用功能 | POP3、FTP、HTTP、Telnet、SMTP  DHCP、TFTP、SNMP、DNS |
| 6 | 表示层 | 数据的格式与表达、加密、压缩 |
| 5 | 会话层 | 建立、管理和终止会话 |
| 4 | 传输层 | 端到端的连接 | TCP、UDP |
| 3 | 网络层 | 分组传输和路由选择 | 三层交换机、路由器  ARP、RARP、IP、ICMP、IGMP |
| 2 | 数据链路层 | 传送以帧为单位的信息 | 网桥、交换机（多端口网桥）、网卡  PPTP、L2TP、SLIP、PPP |
| 1 | 物理层 | 二进制传输 | 中继器、集线器（多端口中继器） |

【备考点拨】

掌握OSI七层模型各个层次对应的主要功能、传输单位、主要物理设备。

### 2.2 TCP/IP协议族（★★★★）

【考法分析】

本知识点主要考查的形式有：给定协议，判断其基于TCP或UDP协议基础；或给定协议，判断其所属层次；或给定端口号，判断其对应协议。

【要点分析】

1、TCP/IP协议层次模型



2、常见协议功能和端口号：

**POP3**：110端口，邮件收取

**SMTP**：25端口，邮件发送

**FTP**：20数据端口/21控制端口，文件传输协议

**HTTP**：80端口，超文本传输协议，网页传输

**DHCP**：67端口，IP地址自动分配

**SNMP**：161端口，简单网络管理协议

**DNS**：53端口，域名解析协议，记录域名与IP的映射关系

**TCP**：可靠的传输层协议

**UDP**：不可靠的传输层协议

**ICMP**：因特网控制协议，PING命令来自该协议

**IGMP**：组播协议

**ARP**：地址解析协议，IP地址转换为MAC地址

**RARP**：反向地址解析协议，MAC地址转IP地址

3、TCP与UDP对比

|  |  |  |
| --- | --- | --- |
|  | TCP | UDP |
| 共同点 | 基于IP协议的传输层协议，可以端口寻址 | |
| 不同点 | 面向连接（连接管理）、三次握手、流量控制、差错校验和重传、IP数据报按序接收不丢失不重复、可靠性强、牺牲通信量、效率低 | 不可靠、无连接、错误检测功能弱，无拥塞控制、无流量控制，有助于提高传输的高速率性。  不对无序IP数据报重新排序、不负责重传、不消除重复IP数据报、不对已收到的数据报进行确认、不负责建立或终止连接，这些由UDP进行通信的应用程序进行处理。 |
| 相关协议 | HTTP、FTP、Telnet、POP3、SMTP | DNS、DHCP、TFTP、SNMP |

4、DHCP

分配方式：固定分配、动态分配和自动分配。

无效地址：169.254.X.X （windows）和 0.0.0.0（Linux）。

5、DNS查询顺序

|  |  |
| --- | --- |
| 浏览器输入域名 | HOSTS→本地DNS缓存→本地DNS服务器→根域名服务器→顶级域名服务器→权限域名服务器。 |
| 主域名服务器接收到域名请求 | 本地DNS缓存→根域名服务器 |

【备考点拨】

1、掌握TCP/IP协议族的分层模型；

2、掌握常见协议的功能和对应端口号；

3、了解TCP与UDP的区别；

4、了解DHCP和DNS的用法。

### 2.3 IP地址与子网划分（★★★★★）

【考法分析】

本知识点的考查形式主要有：给定网络号，求取子网划分的网络号个数和主机号个数；或给定多个子网，求取路由汇聚后的网络号、主机数量等。

【要点分析】

1、相关概念：

（1）IP地址分类：A类地址（网络号8位，0开始），B类地址（网络号16位，10开始），C类地址（网络号24位，110开始），D类地址（组播地址，1110开始），E类地址（保留地址，11110开始）。

2、子网划分：将一个网络划分成多个子网（取部分主机号当子网号，计算主机个数时注意去除全0网络号全1广播地址不能作为主机IP）。

|  |  |
| --- | --- |
| IP | 说明 |
| 127网段 | 回播地址，本地环回地址 |
| 主机号非全0和非全1 | 可作为子网中的主机号使用 |
| 主机号全0地址 | 代表这个网络本身，可作为子网地址使用 |
| 主机号全1地址 | 特定子网的广播地址 |
| 169.254.0.0 | 保留地址，用于DHCP失效(Win) |
| 0.0.0.0 | 保留地址，用于DHCP失效(Linux) |

3、路由汇聚：将多个网络合并成一个大的网络（取部分网络号当主机号）。

4、IPv6：IPv6是设计用于替代现行版本IP协议（IPv4）的下一代IP协议。

（1）IPv6地址长度为128位，地址空间增大了296倍；

（2）灵活的IP报文头部格式。使用一系列固定格式的扩展头部取代了IPv4中可变长度的选项字段。IPv6中选项部分的出现方式也有所变化，使路由器可以简单路过选项而不作任何处理，加快了报文处理速度；

（3）IPv6简化了报文头部格式，字段只有8个，加快报文转发，提高了吞吐量；

（4）提高安全性。身份认证和隐私权是IPv6的关键特性；

（5）支持更多的服务类型；

（6）允许协议继续演变，增加新的功能，使之适应未来技术的发展。

**单播地址（Unicast）**：用于单个接口的标识符。

**任播地址（Anycast）**：泛播地址。一组接口的标识符，IPv4 广播地址。

**组播地址（Multicast）**：IPv6中的组播在功能上与IPv4中的组播类似。

IPv4／IPv6过渡技术有：

（1）双协议栈技术：双栈技术通过节点对IPv4和IPv6双协议栈的支持，从而支持两种业务的共存。

（2）隧道技术：隧道技术通过在IPv4网络中部署隧道，实现在IPv4网络上对IPv6业务的承载，保证业务的共存和过渡。隧道技术包括：6 to 4隧道；6over4隧道；ISATAP隧道。

（3）NAT-PT技术：NAT－PT使用网关设备连接IPv6和IPv4网络。当IPv4和IPv6节点互相访问时，NAT－PT网关实现两种协议的转换翻译和地址的映射。

**IPv6地址压缩**：高位0可省略（多次）；整段0可缩写为1个0（多次）；连续多段0可缩写为双冒号（一次）。

【备考点拨】

1、掌握IP地址分类；

2、掌握子网划分和路由汇聚的计算。

3、了解IPv6地址的特点、分类、合法地址判定、过渡技术。

### 2.4 网络规划与设计（★）

【考法分析】

本知识点的考查形式有：给定一些描述，判断所属设计原则；或给定一些描述，判断所属的设计层次，或给定层次，判断其功能描述正误。

【要点分析】

1、需求分析：网络功能要求；网路的性能要求；网络运行环境的要求；网络的可扩充性和可维护性要求。

2、网络规划原则：实用性原则、开放性原则、先进性原则。

3、网络设计与实施原则：可靠性原则、安全性原则、高效性原则、可扩展原则。

4、层次化网络设计



**核心层：**主要是高速数据交换，实现高速数据传输、出口路由，常用冗余机制。

**接入层：**主要是针对用户端，实现用户接入、计费管理、MAC地址认证、MAC地址过滤、收集用户信息，可以使用集线器代替交换机。

**汇聚层 ：**网络访问策略控制、数据包处理和过滤、策略路由、广播域定义 、寻址。

【备考点拨】

1、了解网络规划原则；

2、掌握核心层、接入层、汇聚层的功能和区别。

### 2.5 网络接入技术（★★）

【考法分析】

本知识点的考查形式有：对3G/4G标准的分类，给定场景描述找到所属的协议。

【要点分析】

1、有线接入

公共交换电话网络PSTN

数字数据网DDN：采用数字传输信道传输数据信号的通信网，优势是网络传输速率高、时延小、质量好、网络透明度高、可支持任何规程、安全可靠，但成本高。

综合业务数字网ISDN：一线通

非对称数字用户线路ADSL：在一对铜双绞线基础上为用户提供上、下行非对称的传输速率。（三种方式：基于以太网--PPPoE，基于ATM—PPPoA，静态IP）

同轴光纤技术HFC

2、无线接入：WiFi、蓝牙、红外、WAPI

3、3G：WCDMA、CMDA2000、TD-SCDMA

4、4G：FDD-LTE--频分、TD-LTE--时分

5、5G：理论峰值1G

【备考点拨】

了解相关协议标准的应用场景。

### 2.6 HTML语言（★★）

【考法分析】

本知识点的考查形式主要是给定一些要求，选择使用的标签，或给定标签，选择对应的描述。

【要点分析】

|  |  |
| --- | --- |
| <a> | 定义锚 |
| <b> | 定义粗体字 |
| <body> | 定义文档的主体 |
| <button> | 定义按钮 |
| <center> | 定义居中文本 |
| <col> | 定义表格中一个或多个列的属性值 |
| <font> | 定义文字的字体、尺寸和颜色 |
| <form> | 定义供用户输入的HTML表单 |
| <frame> | 定义框架集的窗口或框架 |
| <h1> | 定义HTML标题 |
| <hr> | 定义水平线 |
| <html> | 定义HTML文档 |
| <img> | 定义图像 |
| <p> | 定义段落 |
| <script> | 定义客户端脚本 |
| <strong> | 定义强调文本 |
| <table> | 定义表格 |
| <td> | 定义表格中的单元 |
| <tr> | 定义表格中的行 |
| <title> | 定义文档的标题 |

【备考点拨】

1、了解常用的HTML标签。

### 2.7 URL（★）

【考法分析】

本知识点的考查形式有：区分URL各部分含义或判定使用的相应协议，判定域名等级。

【要点分析】

1、URL格式：**协议名://主机名.组名.最高层域名**。例:http://www.baidu.com

2、域名

|  |  |  |  |
| --- | --- | --- | --- |
| 组织模式 | 含义 | 地理模式 | 含义 |
| com | 商业组织 | cn | 中国 |
| edu | 教育机构 | hk | 中国香港 |
| gov | 政府机构 | mo | 中国澳门 |
| mil | 军事部门 | tw | 中国台湾 |
| net | 主要网络支持中心 | us | 美国 |
| org | 上述以外组织 | uk | 英国 |
| int | 国际组织 | jp | 日本 |

【信息安全】

### 2.8 对称加密与非对称加密（★★★）

【考法分析】

本知识点的考查形式有：区分对称加密和非对称加密算法；根据给出的描述判断正误；根据对称加密和非对称加密算法的特点区分选择算法；或根据情景描述，确定在某个阶段使用的秘钥。

【要点分析】

1、对称加密（又称为私人秘钥加密/共享秘钥加密）：加密与解密使用同一秘钥。

特点： 1、加密强度不高，但效率高； 2、密钥分发困难。

（大量明文为了保证加密效率一般使用对称加密）

常见对称密钥加密算法：DES、3DES（三重DES）、 RC-5、IDEA、AES算法。

2、非对称加密（又称为公开密钥加密）：密钥必须成对使用（公钥加密，相应的私钥解密）。

特点：加密速度慢，但强度高。

常见非对称密钥加密算法： RSA、ECC

【备考点拨】

1、掌握对称加密和非对称加密机制的加密过程。

2、记住对称加密和非对称加密的加密算法。

3、了解对称加密和非对称加密的特点和适用情景。

### 2.9 信息摘要与数字签名（★★）

【考法分析】

本知识点的考查形式主要有：根据情景描述，判断某个阶段所使用的秘钥及其所属的对象；或签名、认证相关的概念描述判断正误；或信息摘要相关的概念和算法识别。

【要点分析】

1、数字签名的过程如下图所示（发送者使用自己的私钥对摘要签名，接收者利用发送者的公钥对接收到的摘要进行验证）：



2、常见的摘要算法：MD5(128位)，SHA(160位)。

【备考点拨】

1、掌握数字签名机制的过程；

2、了解摘要的机制；

3、了解常见的摘要算法。

### 2.10 数字证书（★★）

【考法分析】

本知识点常见的考查形式有：给定情景描述，判断数字证书中CA签名的作用；判断数字证书中公钥的作用；数字证书相关的概念描述判断正误。

【要点分析】

1、数字证书的内容包括：CA签名、用户信息（用户名称）、用户公钥等。

2、证书中的CA签名验证数字证书的可靠性。

3、用户公钥：客户端利用证书中的公钥加密，服务器利用自己的私钥解密。

【备考点拨】

1、掌握PKI的基本机制流程。

2、掌握数字证书组成部分的作用。

### 2.11 网络安全协议（★★★）

【考法分析】

本知识点的考查形式主要有：判断协议所属的层次；HTTPS协议、PGP协议、SET协议的描述判断正误。

【要点分析】

1、安全协议分层如图所示：



2、HTTPS协议是HTTP协议与SSL协议的结合，默认端口号443。

3、PGP协议是邮件安全协议。

4、SET协议是电子商务安全协议，涉及电子交易安全。

5、SSH：为建立在应用层基础上的安全协议。SSH 是较可靠，专为远程登录会话和其他网络服务提供安全性的协议。

【备考点拨】

1、掌握协议分层；

2、了解HTTPS、SSL、PGP、SET协议的作用，重点掌握HTTPS协议。

### 2.12 防火墙技术与网络攻击（★★★）

【考法分析】

本知识点的考查形式主要有：对于防火墙技术的描述判断正误；给定一些描述判断所属的网络攻击分类或具体的网络攻击方式（主要有拒绝服务、流量分析、重放等）。

【要点分析】

1、信息安全5个基本要素

机密性：确保信息不暴露给未经授权的实体或进程。（加密）

完整性：只有得到允许的人才能修改数据，并且能够判别出数据是否已经被篡改。（摘要）

可用性：得到授权的实体在需要时可访问数据，即攻击者不能占用所有的资源而阻碍授权者的工作。

可控性：可以控制授权范围内的信息流向及行为方式。（用户权限控制）

可审查性：对出现的信息安全问题提供调查的依据和手段。（审计）

2、常见的攻击行为

**被动攻击**：收集信息为主，破坏保密性。

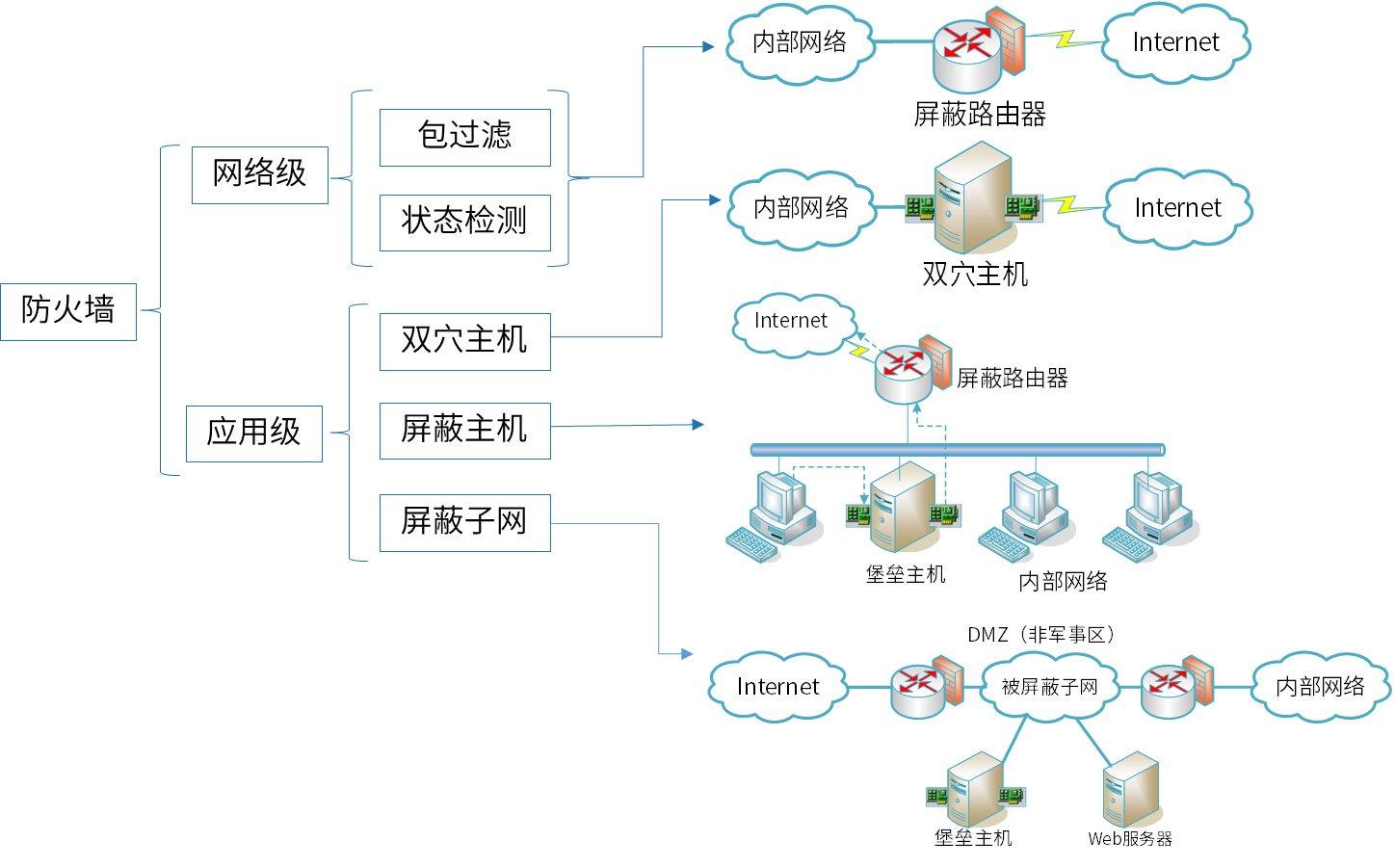
**主动攻击**：主动攻击的类别主要有：中断（破坏可用性），篡改（破坏完整性），伪造（破坏真实性）。

|  |  |  |
| --- | --- | --- |
| 攻击类型 | 攻击名称 | 描述 |
| 被动攻击 | 窃听（网络监听） | 用各种可能的合法或非法的手段窃取系统中的信息资源和敏感信息。 |
| 业务流分析 | 通过对系统进行长期监听，利用统计分析方法对诸如通信频度、通信的信息流向、通信总量的变化等参数进行研究，从而发现有价值的信息和规律。 |
| 非法登录 | 有些资料将这种方式归为被动攻击方式。 |

|  |  |  |
| --- | --- | --- |
| 主动攻击 | 假冒身份 | 通过欺骗通信系统（或用户）达到非法用户冒充成为合法用户，或者特权小的用户冒充成为特权大的用户的目的。黑客大多是采用假冒进行攻击。 |
| 抵赖 | 这是一种来自用户的攻击，比如：否认自己曾经发布过的某条消息、伪造一份对方来信等。 |
| 旁路控制 | 攻击者利用系统的安全缺陷或安全性上的脆弱之处获得非授权的权利或特权。 |
| 重放攻击 | 所截获的某次合法的通信数据拷贝，出于非法的目的而被重新发送。 |
| 拒绝服务（DOS） | 对信息或其他资源的合法访问被无条件地阻止。 |

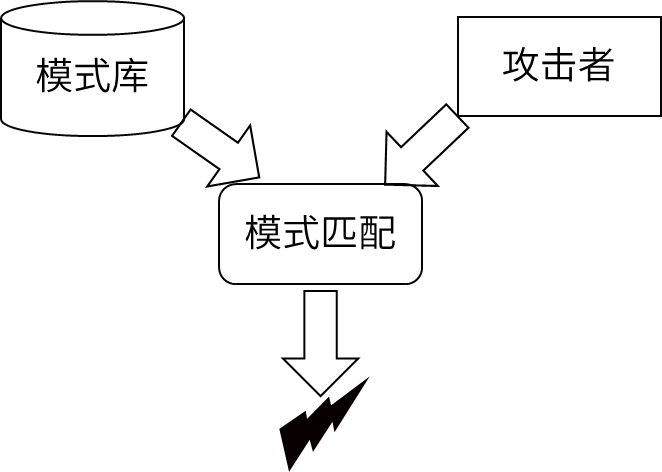
3、常见的防御手段（可以结合使用）

（1）**防火墙技术**：主要了解它的机制是防外不防内，对于DMZ非军事区主要放置应用服务器（如邮件服务器，WEB服务器）。



（2）**漏洞扫描**：入侵者可以利用系统漏洞侵入系统，系统管理员可以通过漏洞扫描技术，及时了解系统存在的安全问题，并采取相应措施来提高系统的安全性。

（3）**入侵检测IDS**：基于数据源的分类——审计功能、记录安全性日志。基于检测方法——异常行为检测。



【备考点拨】

1、了解常见的网络攻击手段和其分类；

2、了解防火墙技术。

### 2.13 计算机病毒与木马（★）

【考法分析】

本知识点的考查形式主要有：给定一些关于病毒、木马的描述，判断正误；或给定一些描述，指出病毒、木马的分类。

【要点分析】

1、常见的病毒、木马命名：

系统病毒（前缀：Win32、PE、W32，如：KCOM——Win32.KCOM）

蠕虫病毒（如：恶鹰——Worm.BBeagle）

木马病毒、黑客病毒（如：QQ消息尾巴木马——Trojan.QQ3344）

脚本病毒（如：红色代码——Script.Redlof）

宏病毒（如：梅丽莎——Macro.Melissa）

后门病毒（如：灰鸽子——Backdoor.Win32.Huigezi）

病毒种植程序病毒（冰河播种者——Dropper.BingHe2.2C）

破坏性程序病毒（杀手命令——Harm.Command.Killer）

玩笑病毒（如：女鬼——Jioke.Grl ghost）

捆绑机病毒（如：捆绑QQ——Binder.QQPass.QQBin）

2、常见的病毒分类：

（1）文件型计算机病毒感染可执行文件（包括EXE和COM文件）。

（2）引导型计算机病毒影响软盘或硬盘的引导扇区。

（3）目录型计算机病毒能够修改硬盘上存储的所有文件的地址。

（4）宏病毒感染的对象是使用某些程序创建的文本文档、数据库、电子表格等文件。

3、病毒的特性：计算机病毒的特性包括隐蔽性、传染性、潜伏性、触发性和破坏性等

【备考点拨】

1、掌握病毒的分类及相关特性；

2、掌握常见病毒的命名，能够区分。

## 3 章节问答

1、计算机网络与信息安全章节的知识内容应该如何复习？

答：

计算机网络与信息安全是软设考试中范围浮动较大的一个章节，不建议全面拓展延伸学习，可以熟悉高频的考点内容，其他低频知识，在做题时进行积累即可。

2、TCP与UDP有什么区别？

答：

TCP和UDP都是传输层协议，可以做端口寻址。前者是可靠的，会有三次握手过程然后建立连接传输数据，可以有差错校验和重传功能，还可以进行流量控制，但是效率相对较差；后者是不可靠的，只负责传输数据，没有建立链接，也没有重传功能等。

3、Telnet为什么与安全的远程连接无关？

答：

Telnet协议是基于TCP的远程连接协议，可以建立远程连接，但是用户名和密码是明文传输的，虽然可靠，但并不安全。与安全的远程连接相关的协议是SSH协议。

4、为什么验证数字网站的真伪需要用CA签名？

答：

验证网站的真伪，其实就是验证数字证书的真伪。类似于验证某个人的身份，一般是检查其身份证。验证是否是官方发放的身份证，需要核实发证机构——公安局，同样，验证数字证书是否是官方发送，也需要核实发证机构——CA，如何证明CA的身份，就需要CA独有的CA签名来证实了。所以数字证书中的CA签名可以验证网站的真伪。

# 第5章 系统开发基础

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）熟悉常见开发模型的特点和适用场景（了解相关关键字，可以区别判断即可）。

（2）了解软件开发方法。

（3）了解需求分析过程及需求的分类。

（4）了解软件设计的任务，以及结构化设计的一些原则，了解模块的内聚性和耦合性。

（5）了解软件测试的分类，了解软件测试的阶段划分，熟悉常用的几种测试方法。（等价类划分、边界值分析、白盒测试覆盖问题）。

（6）掌握McCabe复杂度计算（V(G)=m-n+2）。

（7）区分软件维护的分类和特点（改正性维护，适应性维护，预防性维护，完善性维护）。

（8）了解软件文档的分类（开放文档，产品文档，管理文档）。

（9）了解软件质量保证的质量属性及其子特性归属。

（10）了解软件过程改进的模型（CMMI阶段式和连续式模型）。

（11）了解掌握Gantt图的特点，与pert图区分；

（12）熟悉pert图图示，并熟练掌握相关参数的计算（判断关键路径、项目工期，计算最早开始时间、最晚开始时间、最早完成时间、最晚完成时间，计算总时差）；

（13）了解风险管理相关概念（风险的特性、风险的分类），掌握风险曝光度计算。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 开发模型（★★★★★） | 开发模型 |
| 1 | 软件开发方法（★） | 软件开发方法 |
| 1 | 需求分析（★） | 需求分析 |
| 1 | 软件设计（★★） | 软件设计 |
| 1 | 软件测试（★★★★） | 软件测试 |
| 1 | McCabe复杂度（★★★） | McCabe复杂度 |
| 1 | 软件维护类型（★★★★） | 软件维护类型 |
| 1 | 软件质量保证（★） | 软件质量保证 |
| 1 | 软件过程改进（★★） | 软件过程改进 |
| 1 | Gantt图与Pert图（★★★★） | Gantt图与Pert图 |
| 1 | 风险管理（★★★） | 风险管理 |
| 1 | 【软件设计】数据流图解题技巧 | 数据流图解题技巧 |

## 2 知识点详情

### 2.1 开发模型（★★★★★）

【考法分析】

本知识点的考查形式主要有：给定情景描述或特点描述，指出对应的开发模型；给出特点的开发模型，判断描述的正误；对于统一过程，判断具体任务完成的阶段；对于敏捷开发方法，判断描述正误和一些特点的归属。

【要点分析】

1、**瀑布模型**：瀑布模型是将软件生存周期中的各个活动规定为依线性顺序连接的若干阶段的模型，包括需求分析、设计、编码、运行与维护。

瀑布模型的特点是容易理解，管理成本低，每个阶段都有对应的成果产物，各个阶段有明显的界限划分和顺序要求，一旦发生错误，整个项目推倒重新开始。

适用于需求明确的项目，一般表述为需求明确、或二次开发，或者对于数据处理类型的项目

2、**V模型**：强调测试贯穿项目始终，而不是集中在测试阶段。是一种测试的开发模型。

3、**喷泉模型**：典型的面向对象的模型。特点是迭代、无间隙。会将软件开发划分为多个阶段，但各个阶段无明显界限，并且可以迭代交叉。

4、**原型模型**：典型的原型开发方法模型。适用于需求不明确的场景，可以帮助用户明确需求。

5、**增量模型**：融合了瀑布模型的基本成分和原型实现的迭代特征，可以有多个可用版本的发布，核心功能往往最先完成，在此基础上，每轮迭代会有新的增量发布，核心功能可以得到充分测试。强调每一个增量均发布一个可操作的产品。

6、**螺旋模型**：典型特点是引入了风险分析。结合了瀑布模型和演化模型的优点，最主要的特点在于加入了风险分析。它是由制定计划、风险分析、实施工程、客户评估这一循环组成的，它最初从概念项目开始第一个螺旋。属于面向对象开发模型，强调风险引入。

7、**统一过程**（在软件设计师考试中UP、RUP都指统一过程）：典型特点是用例驱动、以架构为中心、迭代和增量。统一过程把一个项目分为四个不同的阶段：

构思阶段 ：包括用户沟通和计划活动两个方面，强调定义和细化用例，并将其作为主要模型。

细化阶段 ：包括用户沟通和建模活动，重点是创建分析和设计模型，强调类的定义和体系结构的表示。

构建阶段 ：将设计转化为实现，并进行集成和测试。

移交阶段 ：将产品发布给用户进行测试评价，并收集用户的意见，之后再次进行迭代修改产品使之完善

8、**敏捷开发**是一种以人为核心、迭代、循序渐进的开发方法，适用于小团队和小项目，具有小步快跑的思想。常见的敏捷开发方法有极限编程法、水晶法、并列争球法和自适应软件开发方法。

（1）**极限编程**是一种轻量级的开发方法，它提出了四大价值观：沟通、简单、反馈、勇气。五大原则：快速反馈、简单性假设、逐步修改、提倡更改、优质工作。十二个最佳实践：计划游戏、隐喻、小型发布、简单设计、测试先行、重构、结对编程、集体代码所有制、持续集成、每周工作40小时、现场客户和编码标准。

（2）水晶法强调经常交付，认为每一种不同的项目都需要一套不同的策略、约定和方法论。

（3）并列争球法的核心是迭代、增量交付，按照30天进行迭代开发交付可实际运行的软件。

（4）自适应软件开发的核心是三个非线性的，重叠的开发阶段：猜测、合作、学习。

【备考点拨】

1、掌握常见开发模型的特点，能够加以区分；

2、掌握统一过程的4个阶段的任务；

3、了解敏捷开发设计到的原则。

### 2.2 软件开发方法（★）

【考法分析】

本知识点的考查形式主要是描述某种使用情景，要求考生判断适用的开发方法，或给出开发方法，判断描述的正误。

【要点分析】

1、**结构化开发方法**特点：用户至上，严格区分工作阶段，每阶段有任务和结果，

强调系统开发过程的整体性和全局性，系统开发过程工程化，文档资料标准化，

自顶向下，逐步分解（求精）。

2、**原型开发方法**：适用于需求不明确的情况。

3、**面向对象开发方法**：更好的复用性，关键在于建立一个全面、合理、统一的模型，分析、设计、实现三个阶段，界限不明确。

4、面向服务开发方法：面向对象更高标准的抽象。

【备考点拨】

1、掌握不同开发方法的特点并加以区分。

### 2.3 需求分析（★）

【考法分析】

本知识点的考查方式主要有：给定需求描述判断其需求分类；给定概念描述判断正误；指出需求分析的任务、成果产物、工具。

【要点分析】

1、需求分析的任务是解决做什么的问题。

2、需求的分类：

（1）**功能需求**-考虑系统要做什么，在何时做，在何时以及如何修改或升级。

（2）**非功能需求**：考虑软件开发的技术性指标，例如存储容量限制、执行速度、响应时间及吞吐率等。

（3）**设计约束**：除了功能需求和非功能需求以外的需求，例如操作系统限制、开发语言限制等。

3、需求分析的工具有判定表、判定树、数据流图和数据字典。

4、需求分析的产物有：需求规格说明书SRS。

【备考点拨】

1、了解需求分析的任务、工具、产物等有哪些；

2、掌握需求分析的分类。

### 2.4 软件设计（★★）

【考法分析】

本知识点的主要考查形式有：给出软件设计相关描述（概念、原则等）判断正误；或给出一些情景描述指出其内聚类型或耦合类型。

【要点分析】

1、软件设计的任务是解决怎么做的问题。软件设计包括体系结构设计、接口设计、数据设计和过程设计。

过程设计：系统结构部件转换成软件的过程描述。

结构设计：定义软件系统各主要部件之间的关系。

接口设计（人机界面设计）：软件内部，软件和操作系统间以及软件和人之间如何通信。

数据设计：将模型转换成数据结构的定义。好的数据设计将改善程序结构和模块划分，降低过程复杂性。

2、系统设计的基本任务大体上可以分为概要设计和详细设计两个步骤。

（1）概要设计

**设计软件系统总体结构：**基本任务还是采用某种设计方法，将一个复杂的系统按功能划分成模块；确定每个模块的功能；确定模块之间的调用关系；确定模块之间的接口，即模块之间传递的信息；评价模块结构的质量。

**数据结构及数据库设计**：在需求分析阶段对数据的组成、操作约束和数据之间的关系进行了描述，概要设计阶段要加以细化，详细设计阶段则规定具体的实现细节。

**编写概要设计文档**：概要设计说明书、数据库设计说明书、用户手册以及修订测试计划。

**评审**：对设计部分是否完整地实现了需求中规定的功能、性能等要求，设计的可行性，关键的处理以及外部接口定义的正确性、有效性、各部分之间的一致性等都一一进行评审。

（2）详细设计

对每个模块进行详细的算法设计，用某种图形、表格和语言等工具将每个模块处理过程的详细算法描述出来。

对模块内的数据结构进行设计。

对数据库进行物理设计，即确定数据库的物理结构。

其他设计：根据软件系统的类型，还可能需要进行代码设计、输入/输出格式设计，用户界面设计等。

编写详细设计说明书。

评审：对处理过程的算法和数据库的物理结构都要评审。

3、软件设计的原则：**高内聚、低耦合**

（内聚性）

**偶然聚合**：模块完成的动作之间没有任何关系，或者仅仅是一种非常松散的关系。

**逻辑聚合**：模块内部的各个组成在逻辑上具有相似的处理动作，但功能用途上彼此无关。

**时间聚合**：模块内部的各个组成部分所包含的处理动作必须在同一时间内执行。

**过程聚合**：模块内部各个组成部分所要完成的动作虽然没有关系，但必须按特定的次序执行。

**通信聚合**：模块的各个组成部分所完成的动作都使用了同一个数据或产生同一输出数据。

**顺序聚合**：模块内部的各个部分，前一部分处理动作的最后输出是后一部分处理动作的输入。

（耦合性）

**功能聚合**：模块内部各个部分全部属于一个整体，并执行同一功能，且各部分对实现该功能都必不可少

**非直接耦合**：两个模块之间没有直接关系，它们的联系完全是通过主模块的控制和调用来实现的。

**数据耦合**：两个模块彼此间通过数据参数交换信息。

**标记耦合**：一组模块通过参数表传递记录信息，这个记录是某一个数据结构的子结构，而不是简单变量。

**控制耦合**：两个模块彼此间传递的信息中有控制信息。

**外部耦合**：一组模块都访问同一全局简单变量而不是同一全局数据结构，而且不是通过参数表传递该全局变量的信息。

**公共耦合**：两个模块之间通过一个公共的数据区域传递信息。

**内容耦合**：一个模块需要涉及到另一个模块的内部信息。

4、架构风格

数据流风格：批处理序列、管道-过滤器

调用/返回风格：主程序/子程序、面向对象、层次结构（MVC、C/S、B/S）

独立构件风格：进程通信、事件驱动系统（隐式调用）

虚拟机风格：解释器、基于规则的系统

仓库风格：数据库系统、超文本系统、黑板系统

【备考点拨】

1、掌握软件设计的阶段、任务和原则。

2、区分各种内聚类型、耦合类型。

### 2.5 软件测试（★★★★）

【考法分析】

本知识点的考查形式主要有：给定一些描述判断正误；或对于一些具体测试方法判断分类；以及根据要求设计测试用例或判断测试用例的个数。

【要点分析】

1、常见的软件测试方法分类：

（1）静态测试：桌前检查、代码走查、代码审查。

（2）动态测试

黑盒测试：等价类划分、边界值分析、错误推测、因果图。

白盒测试：语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖、路径覆盖。

2、常见的黑盒测试方法：

**等价类划分**：确定无效与有效等价类，设计用例尽可能多的覆盖有效类，设计用例只覆盖一个无效类。

**边界值分析**：处理边界情况时最容易出错，选取的测试数据应该恰好等于、稍小于或稍大于边界值。

3、常见的白盒测试方法：

|  |  |  |
| --- | --- | --- |
|  | 定义 | 特点 |
| 语句覆盖 | 被测试程序中的每条语句至少执行一次。 | 对执行逻辑覆盖很低，一般认为是很弱的逻辑覆盖。 |
| 判定覆盖  （分支覆盖） | 被测程序每个判定表达式至少获得一次“真”值和“假”值（或者程序中每一个判定取“真”分支和取“假”分支至少通过一次。） | 判定覆盖比语句覆盖更强一些。  判定可以是1个条件，也可以是多个条件的组合。 |
| 条件覆盖 | 每一个判定语句中每个逻辑条件的各种可能的值至少满足一次。 | 条件覆盖和判断覆盖没有包含关系。 |
| 判断/条件覆盖 | 判定中每个条件的所有可能取值（真/假）至少出现一次，并使每个判定本身的判定结果（真/假）也至少出现一次。 | 同时满足判定覆盖和条件覆盖 |
| 条件组合覆盖 | 每个判定中的各种可能值的组合都至少出现一次。 | 同时满足判定覆盖、条件覆盖、判定/条件覆盖。 |
| 路径覆盖 | 覆盖被测试程序中所有可能的路径。 |  |
| 基本路径测试 | 每一条独立路径都执行过（即程序中可执行语句至少执行一次）。 | 测试用例个数与环路复杂度一致。判定为关键控制结点，必须出现在基本路径中。 |
| 循环覆盖 | 循环中每个条件都得到验证。 | 注意数组参数可循环验证 |

3、各测试阶段的任务：

（1）**验收测试**：有效性测试、软件配置审查、验收测试。

（2）**系统测试**：恢复测试、安全性测试、强度测试、性能测试、可靠性测试和安装测试。

（3）**集成测试**：模块间的接口和通信。

（4）**单元测试**：模块接口、局部数据结构、边界条件、独立的路径、错误处理。

（5）其他测试：**回归测试**（修改软件后进行的测试，防止引入新的错误）。负载测试（对软件负载能力的测试）。**压力测试**（对软件超负荷条件下运行情况的测试）。

4、测试的基本原则

尽早、不断地进行测试；

程序员避免测试自己设计的程序；

既要选择有效、合理的数据，也要选择无效、不合理的数据；

修改后应进行回归测试；

尚未发现的错误数量与该程序已发现错误数成正比。

【备考点拨】

1、掌握常见的测试方法分类；

2、掌握常见的黑盒测试方法和白盒测试方法。

3、了解软件测试相关的概念，以及一些常见的测试阶段的任务。

### 2.6 McCabe复杂度计算（★★★）

【考法分析】

本知识点考查的主要形式是给定伪代码或程序流程图，计算其McCabe复杂度。

【要点分析】

1、McCabe复杂度计算公式：V(G)=m-n+2，其中m是有向弧的条数，n是结点数。

2、对于伪代码可以先转换为程序流程图，对程序流程图可以最终转换为结点图处理，转换时注意将交点的地方标注为新的结点，以最终的结点图带入公式结算其McCabe复杂度。

【备考点拨】

1、掌握伪代码🡪程序流程图🡪结点图的转换，及McCabe复杂度计算过程。

### 2.7 软件维护类型（★★★★）

【考法分析】

本知识点的考查形式为给定情景描述判断其维护类型。

【要点分析】

1、**更正性维护**：针对真实存在并已经发生的错误进行的维护行为。

2、**预防性维护**：针对真实存在但还未发生的错误进行的维护。

3、**适应性维护**：指使应用软件适应信息技术变化和管理需求变化而进行的修改。企业的外部市场环境和管理需求的不断变化也使得各级管理人员不断提出新的信息需求。

4、**完善性维护**：扩充功能和改善性能而进行的修改。对已有的软件系统增加一些在系统分析和设计阶段中没有规定的功能与性能特征。

【备考点拨】

1、掌握不同维护类型的特点并加以区分。

### 2.8 软件质量保证（★）

【考法分析】

本知识点的考查形式主要是判断某些质量属性与其依从属性的对应关系。

【要点分析】

1、功能性：适合性、准确性、互操作性、安全保密性。

2、可靠性：成熟性、容错性、易恢复性。

3、易用性：易理解性、易学性、易操作性、吸引性。

4、效率：时间特性、资源利用性。

5、维护性：易分析性、稳定性、易测试性、易改变性。

6、可移植性：适应性、易安装性、共存性、易替换性。

【备考点拨】

1、了解软件质量属性及其对应的依从属性。

### 2.9 软件过程改进（★★）

【考法分析】

本知识点考查的主要形式是给定描述判断其所属的CMMI的等级或层次；给定等级或层次，判断其任务、相关描述的正误。

【要点分析】

1、CMMI（Capability Maturity Model Integration，能力成熟度模型集成）

**初始级**：杂乱无章，甚至混乱，几乎没有明确定义的步骤，项目的成功完全依赖个人的努力和英雄式核心人物的作用。

**可重复级**：建立了基本的项目管理过程和实践来跟踪项目费用、进度和功能特性，有必要的过程准则来重复以前在同类项目中的成功。

**已定义级**：管理和工程两方面的软件过程已经文档化、标准化，并综合成整个软件开发组织的标准过程。

**已管理级**：制定了软件过程和产品质量的详细度量标准。

**优化级**：加强了定量分析，通过来自过程质量反馈和来自新观念、新技术的反馈使过程能不断持续地改进。

2、CMMI 阶段式模型：

**初始的**：过程不可预测且缺乏控制；

**已管理的**：过程为项目服务；

**已定义的**：过程为组织服务；

**定量管理的**：过程已度量和控制；

**优化的**：集中于过程改进。

3、CMMI连续式模型：

**CL0（未完成的）**：过程域未执行或未得到CL1中定义的所有目标。

**CL1（已执行的）**：其共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品，以实现支持过程域的特定目标。

**CL2（已管理的）**：其共性目标是集中于已管理的过程的制度化。根据组织级政策规定过程的运作将使用哪个过程，项目遵循已文档化的计划和过程描述，所有正在工作的人都有权使用足够的资源，所有工作任务和工作产品都被监控、控制、和评审。

**CL3（已定义级的）**：其共性目标集中于已定义的过程的制度化。过程是按照组织的裁剪指南从组织的标准过程中裁剪得到的，还必须收集过程资产和过程的度量，并用于将来对过程的改进。

**CL4（定量管理的）**：其共性目标集中于可定量管理的过程的制度化。使用测量和质量保证来控制和改进过程域，建立和使用关于质量和过程执行的质量目标作为管理准则。

**CL5（优化的）**：使用量化（统计学）手段改变和优化过程域，以满足客户的改变和持续改进计划中的过程域的功效。

3、CMM模型

初始级：杂乱无章，甚至混乱，几乎没有明确定义的步骤，项目的成功完全依赖个人的努力和英雄式核心人物的作用。

可重复级：建立了基本的项目管理过程和实践来跟踪项目费用、进度和功能特性，有必要的过程准则来重复以前在同类项目中的成功。

已定义级：管理和工程两方面的软件过程已经文档化、标准化，并综合成整个软件开发组织的标准过程。

已管理级：制定了软件过程和产品质量的详细度量标准。

优化级：加强了定量分析，通过来自过程质量反馈和来自新观念、新技术的反馈使过程能不断持续地改进。

【备考点拨】

1、了解软件过程改进模型各个层级、层次的特点，能够加以区分。

【项目管理】

### 2.10 Gantt图与Pert图（★★★★）

【考法分析】

本知识点的考查形式主要有：判断对于Gantt图和Pert图的描述是否正确；根据图示找到Pert图的关键路径，计算其项目工期，计算某些活动的松弛时间，判断某些活动的最早/最晚开始或完成时间。

【要点分析】

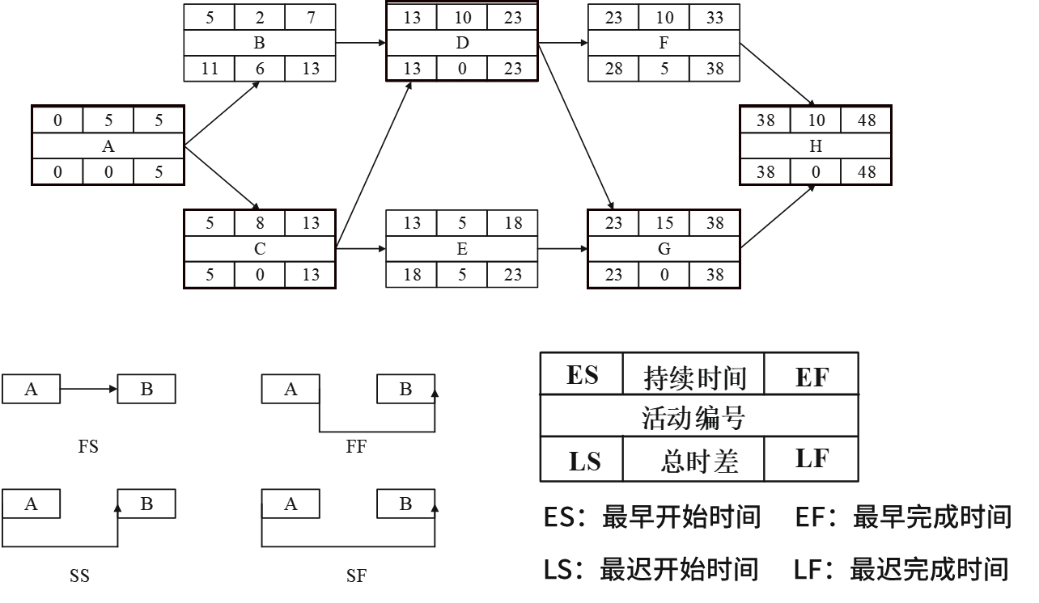
1、甘特图能够清晰描述每个任务的开始/结束时间及各任务之间的并行性，也可以动态地反映项目的开发进展情况，但难以反映多个任务之间存在的逻辑关系；PERT利用项目的网络图和各活动所需时间的估计值（通过加权平均得到的）去计算项目总时间，强调任务之间的先后关系，但不能反映任务之间的并行性，以及项目的当前进展情况。

2、关键路径法是图中源点至汇点的最长路径，关键路径的时间称之为项目工期，也表述为项目完成所需的最少时间。

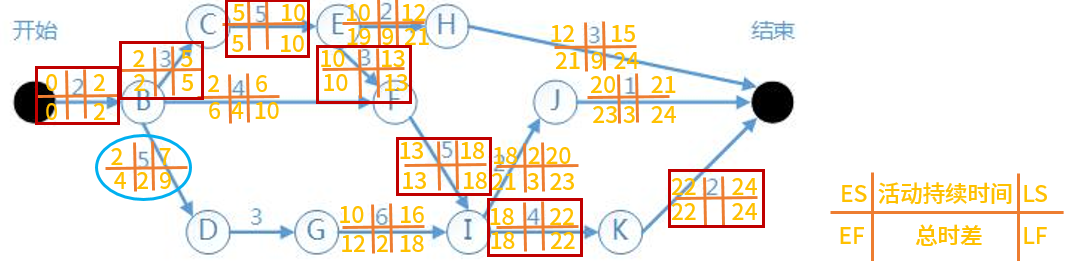
3、总时差：在不延误总工期的前提下，该活动的机动时间。一般在图中，以最晚结束时间减去最早结束时间求取，或以最晚开始时间减去最早开始时间求取。

4、对于网络图我们一般采用关键路径分析法处理，关键路径法是利用进度模型时使用的一种进度网络分析技术。沿着项目进度网络路线进行正向与反向分析，从而计算出所有计划活动理论上的最早开始与完成日期、最迟开始与完成日期，不考虑任何资源限制。

5、单代号网络图：结点表示活动，箭线表示活动与活动间的依赖关系。



6、双代号网络图：结点表示里程碑，箭线表示活动。



【备考点拨】

1、掌握Gantt图与Pert图相关的概念；

2、掌握Pert图分析的方法，能够标注出单代号网络图和双代号网络图中的最早开始/结束数据，和最晚开始/结束时间，并以此找出图中的关键路径，掌握相关参数的计算方法。

### 2.11 风险管理（★★★）

【考法分析】

本知识点的考查形式，主要有：对风险相关概念描述判断正误；计算风险曝光度。

【要点分析】

1、风险的特性：具有不确定性，可能会造成损失。

2、风险的类别：项目风险涉及到各种形式的预算、进度、人员、资源以及客户相关的问题，并且可能导致项目损失。技术风险涉及到技术相关的可能会导致项目损失的问题。商业风险与市场因素相关。社会风险涉及到政策、法规等因素。

3、风险暴露又称风险曝光度，测量的是资产的整个安全性风险，它将表示实际损失的可能性与表示大量可能损失的资讯结合到单一数字评估中。在形式最简单的定量性风险分析中，风险曝光度可透过将风险可能性及影响相乘算出。

风险曝光度（RiskExposure）=错误出现率（风险出现率）×错误造成损失（风险损失）。

【备考点拨】

1、掌握风险相关的一些概念；

2、掌握风险曝光度的计算。

### 2.12 沟通管理（★）

【考法分析】

本知识点主要考查沟通路径的计算。

【要点分析】

1、有主程序员：n个成员小组，1个主程序员，普通程序员只需要与主程序员沟通。

沟通路径：n-1。

2、无主程序员：n个成员的项目小组，相互之间都可以沟通。

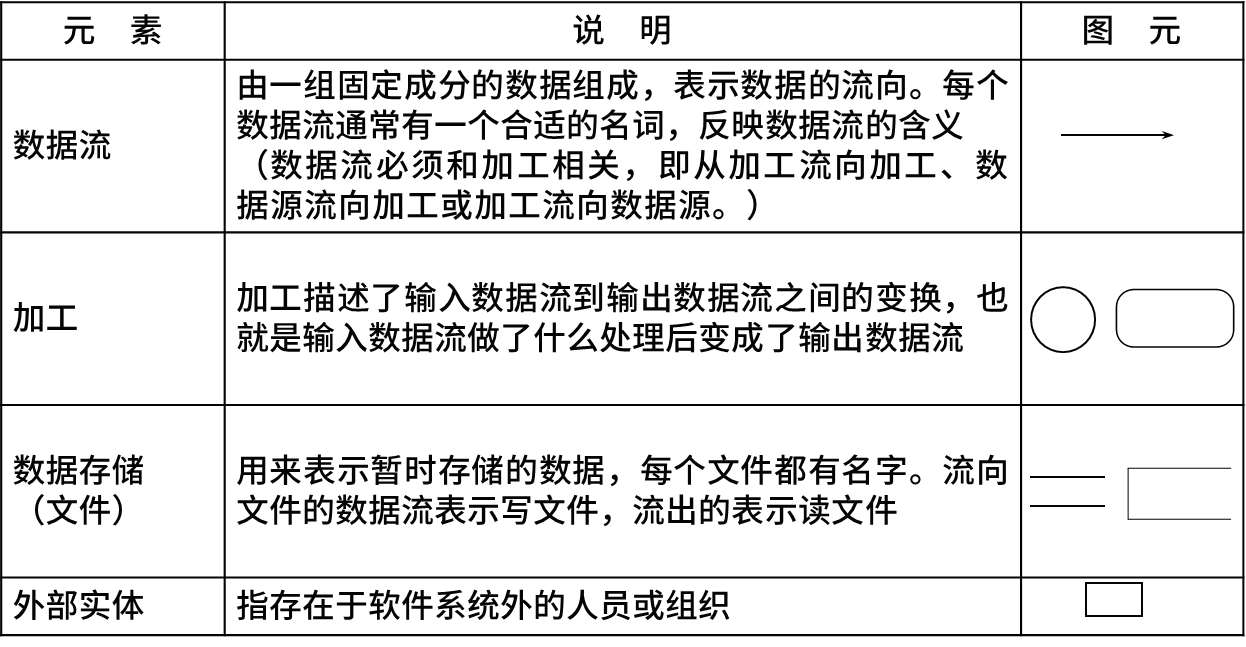
沟通路径：n(n-1)/2。

【备考点拨】

1、掌握不同形式下沟通路径的计算。

### 2.13 【软件设计】数据流图解题技巧

1、数据流图图示



2、填空技巧

（1）补充实体

实体可能是：

人物角色：如客户、管理员、主管、经理、老师、学生

组织机构：如银行、供应商、募捐机构

外部系统：如银行系统、工资系统、后台数据库（当要开发的是中间件时）

（2）补充存储

存储的文字方面特征：“\*\*文件”“\*\*表”“\*\*库” “\*\*清单” “\*\*档案”本考点的基本考法是与内存地址计算、IP地址计算结合考查。

（3）补充加工名

加工是用于处理数据流的，所以要补充加工名，可以把该加工涉及到的数据流，在说明中标识出来，再在数据流名称所在的句子中，找“动词+名词”的结构，分析是否可作为加工。

“动词+名词”如：生成报告，发出通知，批改作业，记录分数，当然这只是普遍情况，也有例外，如物流跟踪、用户管理。

（4）补充数据流

数据平衡原则：

顶层图与0层图对比，是否有顶层图有，但0层图无的数据流，或反之。

检查图中每个加工，是否存在只有入没有出，或只有出没有入，或根据输入的数据无法产生对应的输出的情况。

按题目说明与图进行匹配：

说明中的每一句话，都能与图中有对应关系，当把说明中的实体与数据流标识出来之后，容易缩小对应范围，找出纰漏。

3、主观题分析

（1）数据字典

|  |  |  |
| --- | --- | --- |
| 符 号 | 含 义 | 举 例 说 明 |
| = | 被定义为 |  |
| + | 与 | x=a+b，表示x由a和b组成 |
| […，…]或[…|…] | 或 | x=[a，b]，x=[a|b]，表示x由a或由b组成 |
| {…} | 重复 | x={a}，表示x由0个或多个a组成 |
| (…) | 可选 | x=(a)，表示a可在x中出现，也可以不出现 |

（2）数据流图常见的3种错误：

加工只有输入没有输出，称之为“黑洞”；

加工只有输出没有输入，称之为“奇迹”；

加工中输入不足以产生输出，称之为“灰洞”。

（3）子图与父图保持平衡

父图与子图之间平衡是指任何一张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/输出数据了保持一致。如果父图中某个加工的一条数据流对应于子图中的几条数据流，而子图中组成这些数据流的数据项全体正好等于父图中的这条数据流，那么它们仍然是平衡的。

（4）对加工进行结构化描述

结构化语言是一种介于自然语言和形式化语言之间的半形式化语言，是自然语言的一个受限子集。

外层：用来描述控制结构，采用顺序、选择和重复3种基本结构。

①顺序结构，一组祈使语句、选择语句、重复语句的顺序排列。

②选择结构，一般用IF-THEN-ELSE-ENDIF、CASE-OF-ENDCASE等关键词。

③重复结构，一般用DO-WHILE-ENDDO、REPEAT-UNTIL等关键词。

④内层：一般采用祈使语句的自然语言短语，使用数据字典中的名词和游戏的自定义词，其动词含义要具体，尽量不用形容词和副词来修饰，还可使用一些简单的算法运算和逻辑运算符号。

## 3 章节问答

1、瀑布模型适用于需求明确的项目，但需求又具有渐进明晰性，考试中需求明确是如何体现的？

答：

一般题干描述出现二次开发、行业经验等都属于需求明确的描述。

2、对于软件设计中模块设计的过程需要遵循的原则，除了“高内聚低耦合”还有哪些？

答：

对于模块设计除了“高内聚，低耦合”，还要适当考虑权衡适量的原则，模块大小适中，适宜的系统深度和宽度比例，尽可能减少调用的深度，适度控制模块的扇入扇出，模块的作用域应该在模块之内等。

3、项目工期（项目的最短完工时间）为什么由最长路径的时间决定而不是最短路径的时间决定？

答：

对于一个项目的完成，应该是所有活动全部完成才算是正式完工，如果选择最短路径，此时在当前时间范围内，会有一些项目活动并没有完成，此时项目不能提交给用户，也没有完工，而选择最长路径，在这个时间范围内，其他活动一定可以完成，所以整个项目可以正式完工，因此需要最长路径的时间作为整个项目的最短完工时间，这个路径叫作关键路径（关键路径可以有多条，项目工期只有1个值）。

# 第6章 面向对象

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）熟悉常见的面向对象概念；

（2）熟悉面向对象设计原则；

（3）了解UML分类和常见的UML图示。

（4）掌握设计模式：首先辨识中英文名称，其次掌握设计模式分类，然后了解各设计模式的使用场景。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 面向对象的概念（★★★★★） | 面向对象的概念 |
| 1 | UML（★★★★★） | UML |
| 1 | 设计模式（★★★★★） | 设计模式 |

## 2 知识点详情

### 2.1 面向对象的概念（★★★★★）

【考法分析】

本知识点的考查形式主要是给定相关的一些概念描述判断正误；或根据描述指出对应的概念。

【要点分析】

1、基本概念：

**对象**：属性（数据）+方法（操作）+对象ID

**封装**：隐藏对象的属性和实现细节，仅对外公开接口（信息隐藏技术）。

**类**（实体类/控制类/边界类）

**接口**：一种特殊的类，他只有方法定义没有实现。

**继承与泛化**：复用机制。

**重置/覆盖**（Overriding）：在子类中重新定义父类中已经定义的方法。

**动态绑定**：根据接收对象的具体情况将请求的操作与实现的方法进行连接（运行时绑定）。

**多态**：不同对象收到同样的消息产生不同的结果，多态实质上是将子类的指针对象或者引用对象传递给父类指针对象后，通过这个父类指针对象调用的函数（此函数在父类中声明为虚函数，且在各个子类中重写这个函数），不是父类中定义的，而是传递进来的子类对象中重写的函数。（软设考试中对于多态分类只出现过过载多态-过载多态：同一个名字在不同的上下文中所代表的含义不同。）。

**重载**：一个类可以有多个同名而参数类型不同的方法。

**类属类**：类的模板。

**消息和消息通信**：对象之间进行通信的一种构造叫作消息。消息是异步通信的（消息传递：接收到信息的对象经过解释，然后予以响应）

2、面向对象设计原则：

（1）**单一职责原则**：设计目的单一的类。

（2）**开放-封闭原则**：对扩展开放，对修改封闭。

（3）李氏（Liskov）替换原则：子类可以替换父类。

（4）**依赖倒置原则**：要依赖于抽象，而不是具体实现；针对接口编程，不要针对实现编程。

（5）**接口隔离原则**：使用多个专门的接口比使用单一的总接口要好。

（6）**组合重用原则**：要尽量使用组合，而不是继承关系达到重用目的。

（7）**迪米特（Demeter）原则（最少知识法则）**：一个对象应当对其他对象有尽可能少的了解。

4、面向对象其他设计原则

重用发布等价原则：重用的粒度就是发布的粒度

共同封闭原则：包中的所有类对于同一性质的变化应该是共同封闭的。一个变化若对一个包产生影响，则将对该包里的所有类产生影响，而对于其他的包不造成任何影响。

共同重用原则：一个包里的所有类应该是共同重用的。如果重用了包里的一个类，那么就要重用包中的所有类。

无环依赖原则：在包的依赖关系图中不允许存在环，即包之间的结构必须是一个直接的无环图形。

稳定依赖原则：朝着稳定的方向进行依赖。

稳定抽象原则：包的抽象程度应该和其稳定程度一致。

4、面向对象开发过程

面向对象分析：认定对象（名词）；组织对象（抽象成类）；对象间的相互作用；基于对象的操作。

面向对象设计：识别类及对象；定义属性；定义服务；识别关系；识别包。

面向对象开发：程序设计范型；选择一种OOPL。

面向对象测试：算法层；类层；模板层；系统层。

【备考点拨】

1、掌握面向对象相关的基本概念。

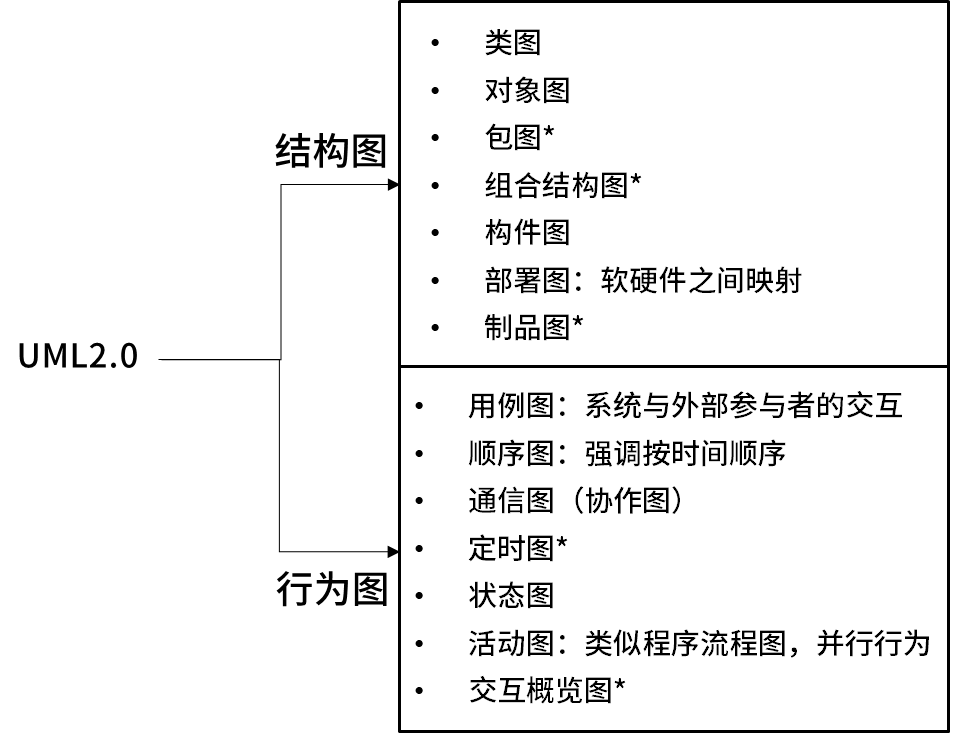
### 2.2 UML（★★★★★）

【考法分析】

本知识点的主要考查形式有：给定图示判断具体属于哪种UML图；给定UML图，判断属于哪一类（结构图或行为图）；给定UML图，指出相关的一些参数；给出一些关于UML的描述，判断正误。

【要点分析】

1、UML图分类：



（早期对于用例图也有放到结构图一类）

2、用例图：用例图描述一组用例、参与者及它们之间的关系。



用例之间的关系：

**包含关系**：其中这个提取出来的公共用例称为抽象用例，而把原始用例称为基本用例或基础用例：当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。

**扩展关系**：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。

**泛化关系**：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。在用例的泛化关系中，子用例是父用例的一种特殊形式，子用例继承了父用例所有的结构、行为和关系。

3、类图（class diagram）：类图描述一组类、接口、协作和它们之间的关系。在OO系统的建模中，最常见的图就是类图。类图给出了系统的静态设计视图，活动类的类图给出了系统的静态进程视图。

对象图（object diagram）：对象图描述一组对象及它们之间的关系。对象图描述了在类图中所建立的事物实例的静态快照。和类图一样，这些图给出系统的静态设计视图或静态进程视图，但它们是从真实案例或原型案例的角度建立的。



类之间的关系：

**依赖关系**：一个事物发生变化影响另一个事物。

**泛化关系**：特殊/一般关系

**关联关系**：描述了一组链，链是对象之间的连接。

**聚合关系**：整体与部分生命周期不同。

**组合关系**：整体与部分生命周期相同。

**实现关系**：接口与类之间的关系

4、顺序图（sequence diagram，序列图）。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。交互图专注于系统的动态视图。顺序图是强调消息的时间次序的交互图。



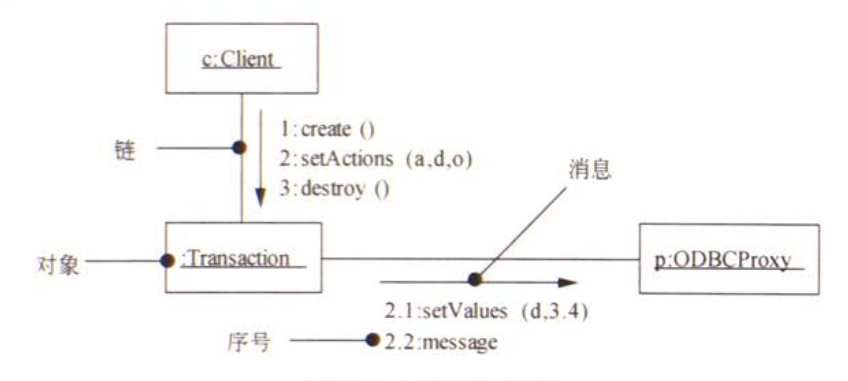
5、活动图（activity diagram）。活动图将进程或其他计算结构展示为计算内部一步步的控制流和数据流。活动图专注于系统的动态视图。它对系统的功能建模和业务流程建模特别重要，并强调对象间的控制流程。



6、状态图（state diagram）。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。它对于接口、类或协作的行为建模尤为重要，而且它强调事件导致的对象行为，这非常有助于对反应式系统建模。



7、通信图（communication diagram）。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。顺序图和通信图表达了类似的基本概念，但它们所强调的概念不同，顺序图强调的是时序，通信图强调的是对象之间的组织结构（关系）。



8、构件图（component diagram）。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。构件图用于表示系统的静态设计实现视图。对于由小的部件构建大的系统来说，构件图是很重要的。构件图是类图的变体。



9、部署图（deployment diagram）。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。部署图给出了架构的静态部署视图，通常一个节点包含一个或多个部署图。



【备考点拨】

1、掌握各类UML图的特点和适用情景，能够加以区分。对一些特殊的图的特殊部分能够辨认区分。

### 2.3 【软件设计】UML建模解题技巧

【考法分析】

UML建模是软件设计师考试下午题的第3题，考查形式为：给定UML图，抠出其中一部分内容，要求大家进行填空，并且可能补充其他的一些题型进行考查，比如对于类图、用例图中关系的考查，结合设计模式进行考查等。

【要点分析】

1、用例图常见考查形式：

（1）补充参与者：参与者一般为外部实体，可以是人，也可以是外部系统。

（2）补充用例：根据题干描述补充缺失的用例，在补充过程中需要参照用例图中用例间的关系。（用例名从题干中直接获取或提炼）

（3）分析用例间的关系

（4）其他：可能会出现其他题型，比如添加新的用例等。

2、类图/对象图常见考查形式（常与其他UML图结合考查）：

（1）补充类名/对象名：一般为名词，来源于题干描述。

（2）补充多重度：多重度体现类与类或对象与对象之间的对应关系，有0…1，1..1，1…\*/n，\*…\*/m…n。

（3）分析类与类或对象与对象之间的关系。

3、常见考查形式：

（1）补充类名/对象名

（2）补充消息

4、常见考查形式：

（1）补充动作名称

（2）补充监护表达式

（3）分析并发关系

（4）活动图与状态图对比，活动图与进程图关系。

5、状态图常见考查形式：

（1）补充状态

（2）补充触发事件、监护条件、动作

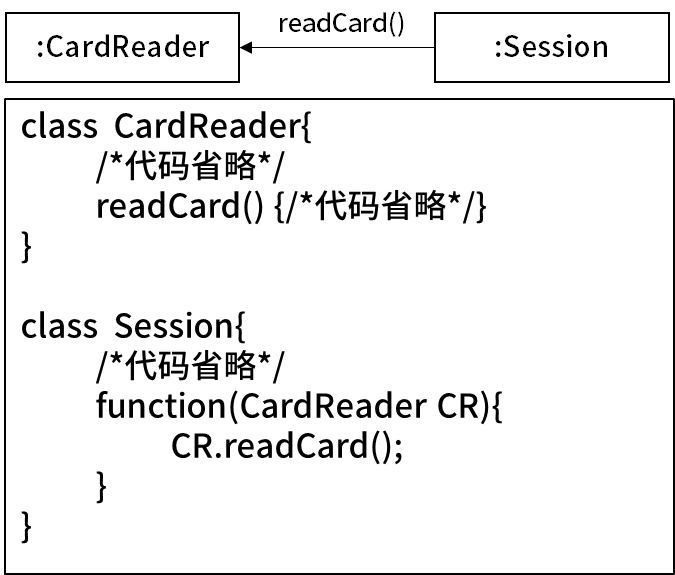
（3）与状态模式结合考查

6、通信图常见考查形式：

（1）补充对象

（2）补充消息

7、消息的所属判断：箭头上的消息属于箭头指向的对象/类。



### 2.4 设计模式（★★★★★）

【考法分析】

1、本知识点的主要考查形式有： 根据图示题干描述和图示指出对应的设计模式（中文或英文形式），指出该设计模式的分类，或该设计模式的适用场景。

【要点分析】

1、设计模式的分类：



2、设计模式应用场景和记忆关键字：

（1）创建型模式

|  |  |  |
| --- | --- | --- |
| 设计模式名称 | 简要说明 | 速记关键字 |
| Factory Method  工厂方法模式 | 定义一个创建对象的接口，但由子类决定需要实例化哪一个类。 工厂方法使得子类实例化的过程推迟 | 动态生产对象 |
| Abstract Factory  抽象工厂模式 | 提供一个接口，可以创建一系列相关或相互依赖的对象，而无需指定它们具体的类 | 生产成系列对象 |
| Builder  构建器模式 | 将一个复杂类的表示与其构造相分离，使得相同的构建过程能够得出不同的表示 | 复杂对象构造 |
| Prototype  原型模式 | 用原型实例指定创建对象的类型，并且通过拷贝这个原型来创建新的对象 | 克隆对象 |
| Singleton  单例模式 | 保证一个类只有一个实例，并提供一个访问它的全局访问点 | 单实例 |

（2）结构型模式

|  |  |  |
| --- | --- | --- |
| 设计模式名称 | 简要说明 | 速记关键字 |
| Adapter  适配器模式 | 将一个类的接口转换成用户希望得到的另一种接口。它使原本不相容的接口得以协同工作 | 转换接口 |
| Bridge  桥接模式 | 将类的抽象部分和它的实现部分分离开来，使它们可以独立地变化 | 继承树拆分 |
| Composite  组合模式 | 将对象组合成树型结构以表示“整体-部分”的层次结构，使得用户对单个对象和组合对象的使用具有一致性 | 树形目录结构 |
| Decorator  装饰模式 | 动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代，比派生一个子类更加灵活 | 动态附加职责 |
| Facade  外观模式 | 定义一个高层接口，为子系统中的一组接口提供一个一致的外观，从而简化了该子系统的使用 | 对外统一接口 |
| Flyweight  享元模式 | 提供支持大量细粒度对象共享的有效方法 | 汉字编码 |
| Proxy  代理模式 | 为其他对象提供一种代理以控制这个对象的访问 | 快捷方式 |

（3）行为型模式

|  |  |  |
| --- | --- | --- |
| 设计模式名称 | 简要说明 | 速记关键字 |
| Chain of  Responsibility  职责链模式 | 通过给多个对象处理请求的机会，减少请求的发送者与接收者之间的耦合。将接收对象链接起来，在链中传递请求，直到有一个对象处理这个请求 | 传递职责 |
| Command  命令模式 | 将一个请求封装为一个对象，从而可用不同的请求对客户进行参数化，将请求排队或记录请求日志，支持可撤销的操作 | 日志记录，可撤销 |
| Interpreter  解释器模式 | 给定一种语言，定义它的文法表示，并定义一个解释器，该解释器用来根据文法表示来解释语言中的句子 | 虚拟机的机制 |
| Iterator  迭代器模式 | 提供一种方法来顺序访问一个聚合对象中的各个元素，而不需要暴露该对象的内部表示 | 数据集 |
| Mediator  中介者模式 | 用一个中介对象来封装一系列的对象交互。它使各对象不需要显式地相互调用，从而达到低耦合，还可以独立地改变对象间的交互 | 不直接引用 |
| Memento  备忘录模式 | 在不破坏封装性的前提下，捕获一个对象的内部状态，并在该对象之外保存这个状态，从而可以在以后将该对象恢复到原先保存的状态 | 游戏存档 |
| Observer  观察者模式 | 定义对象间的一种一对多的依赖关系，当一个对象的状态发生改变时，所有依赖于它的对象都得到通知并自动更新 | 联动 |
| State  状态模式 | 允许一个对象在其内部状态改变时改变它的行为 | 状态变成类 |
| Strategy  策略模式 | 定义一系列算法，把它们一个个封装起来，并且使它们之间可互相替换，从而让算法可以独立于使用它的用户而变化 | 多方案切换 |
| Template Method  模板方法模式 | 定义一个操作中的算法骨架，而将一些步骤延迟到子类中，使得子类可以不改变一个算法的结构即可重新定义算法的某些特定步骤 | 框架 |
| Visitor  访问者模式 | 表示一个作用于某对象结构中的各元素的操作，使得在不改变各元素的类的前提下定义作用于这些元素的新操作 | 数据与操作分离 |

【备考点拨】

1、掌握23种设计模式的中英文；

2、掌握23种设计模式的分类；

3、根据关键字，掌握23种设计模式的应用场景、相关描述；

4、了解23种设计模式对应的UML图示。

### 2.5 【软件设计】面向对象程序设计（JAVA）解题技巧

1、了解类与类（接口）之间的关系——接口是一种特殊的类

（1）接口的概念：接口只是说明类应该做什么，但并不指定应该如何去做。在实际开发过程中，通过类来实现接口。接口一般只有方法名，没有方法体。实现接口就是让其既有方法名又有方法体。

（2）接口的声明：接口用关键字“interface”来声明。接口的形式跟类很相似。例1格式如下：

interface [接口名]{

…

}

（3）接口的实现

实现接口的类表示例2如下：（接例1接口的声明）

class [类名] implements [接口名]{

…

}

（4）特殊的接口方法：接口中的方法只有方法名没有方法体。

（注意：没有花括号）例3如下：

interface [接口名]{

[返回类型void/String/…] [方法名] （参数列表）；

}

（5）接口类与其实现类的对应关系：接口的用处就是让类通过实现它来执行一定的功能。因此实现接口的类，要实现接口的类。对于实现类或接口类有方法或属性的缺失，则可以根据二者的对应关系进行补充。一般实现类的方法是对接口类方法的具体实现。

2、继承

（1）继承的概念：继承，就是在已有类（父类）的基础上进行扩展，从而产生新的类（子类）。父类拥有自己的属性和方法，这些子类都可以继承。子类除了拥有父类的属性和方法外，还可以有自己的特性。

（2）继承的表示：继承通过关键字“extends”表示。结构例4如下：

class [子类名] extends [父类名]{

…

}

（3）父类与子类之间的对应关系：子类的属性和方法有部分是从父类继承而来，一般而言，父类和子类之间同名的属性和方法具有一定的对应关系，子类也可以对父类的方法进行重写。

（4）super和this的使用

在继承关系中为了区分父类和子类，会用super代替父类指针，this代替子类指针。举例如下：

class [父类名] {

[属性1]；

[父类名]（[属性1]）{ 属性1=1； }

}

class [子类名]{

[属性2]；

[子类名]（[属性1],[属性2]）{

super（[属性1]）； //这里的super（[属性1]）时父类带属性1参数的构造函数

this.[属性2]=[属性2]； //这里的this是当前类指针

//加上this和super可以区分同名的属性或方法。子类可以利用super指针调用父类的属性和方法。

}

}

3、了解类的结构——属性和构造函数

（1）默认构造函数和非默认的构造函数：对类实例化创建对象时，系统会调用构造函数对其所属成员进行初始化。

在JAVA中，一般以与类名同名的方法作为构造函数。在实际开发过程中，系统会默认不带参数的构造函数，如果需要带参数的构造函数，需要明确写出该构造函数。构造函数结构例5如下：

class [类1]{

[属性1]；

//[类1](){ } //默认无参构造函数

[类1](属性1 形参1){

this.属性1=形参1； //一般这里的形参命名与属性名一致

}

}

（2）继承中的构造函数：实际上，在创建子类对象时，会先执行其父类的构造函数，然后执行子类的构造函数，最后完成对象的创建。即创建子类对象时，先调用父类构造函数，初始化继承自父类的成员，随后调用子类构造函数，初始化子类的成员。例6如下：

class [父类名]{

[属性1]；

[父类名]（）{

[属性1]=10；

System.out.println(“这是父类构造函数”);

}

}

class [子类名]{

[属性2]；

[子类名]（）{

[属性2]=20；

System.out.println(“这是子类构造函数”);

}

}

//主函数测试

public class test{

public static void main(String [] args)

{

[子类名] [子类对象名]= new [子类名] （）；

System.out.println([子类对象名] .[属性1]+“ ”+[子类对象名] .[属性2]);

}

}

//这里的输出结构应该是：

// 这是父类构造函数

// 这是子类构造函数

// 10 20

（3）get方法和set方法

在JAVA中会有对类中属性的get和set方法。在实际开发过程中，会对类的属性设定私有private限定，只有本类中的方法可以访问，拒绝其他类的访问。同时，会在本类中设定get()和set()方法，对相关属性进行操作。这些在开发工具中，有时候会默认自动给出。结构如下所示：

class [类名]{

private [数据类型] [属性1]；

public [数据返回类型] get属性1（）{

return [属性1]；

}

public void set属性1（[数据类型] [形参]）{

this.[属性1]=[形参]； //这里的形参名一般与属性名一致

}

}

（4）其他函数：类中函数成员格式如下所示：

[访问控制符] [返回类型] [方法名] (形参1类型 形参名1，形参2类型 形参名2，…){ … }

访问控制符可以为：private/public/default/protected/abstract

返回类型为该方法返回数据的数据类型。

返回语句 return [返回参数]

如果返回类型为void，即返回类型为空时，不需要return语句。

4、访问控制符/类修饰符

private

（1）用private修饰的成员变量与成员方法只能在类的内部被访问，类的外部不能访问。

public

（1）用public修饰数据表示所有的类都可以访问。

（2）用public修饰类，也表示所有类中可以访问。

default

（1）用default修饰成员数据表示只有用一个包里的类才能访问。

（2）用default修饰类也表示只有一个包里的类才能访问。当在类前不加任何控制符时，默认就是default。

protected

（1）用protected修饰成员数据表示不仅同一个包中的类可以访问，位于其他包中的子类也可以访问。

abstract

（1）abstract可以用来修饰方法或类。具有一个或多个抽象方法的类，本身需要被定义为抽象类。抽象类不仅可以有抽象方法，还可以有具体方法。

（2）抽象类可以被继承，如果其子类没有实现抽象类全部的抽象方法，则子类也是抽象类，需要用abstract修饰。

（3）含有抽象方法的一定是抽象类，但抽象类可以没有抽象方法。抽象类不能实例化，不能new一个对象，但可以声明一个抽象类的变量指向具体的子类对象。

5、了解对象的实例化

（1）new一个对象

在主函数中，需要对类进行实例化才能够调用。而我们对类的实例化，一般用的是new。结构如下：

[类名] [对象名] = new [类名]（参数列表）；

（2）与超类相关的实例化

//创建子类对象并转到父类

[父类名] [对象名] = new [子类名](参数列表)；

//此时以对象名调用某方法（子类从父类直接继承而来的方法），虽然将父类对象句柄指向了子类对象，实际上操作的还是子类对象，只不过将对象句柄声明为父类的数据类型，此时编译器根据实际情况选择了子类的函数。

6、了解方法调用的格式：

[对象名] . [方法名] ;

7、了解特殊的数据结构接口

List（表单）

（1）将元素放到指定集合的结尾

[表单名]. add([元素名])；

（2）将集合2放到指定集合1的指定位置

[表单名1].addAll([位置序号]，[表单名2])；

（3）清除集合

[表单名].clear()；

Iterator（迭代器）--用迭代器访问表单

## 3 章节问答

1、设计模式要求掌握到什么程度？

答：

设计模式有一定难度。对于设计模式要求掌握如下内容：1、设计模型三种类型的定位；2、设计模式分类；3、各设计模式的应用场景及特点，可见上文列出的关键字表格进行记忆。设计模式对应的UML类图可自行拓展学习。

2、如何区分泛化关系？

答：

泛化关系一般可以体现为is-a或者has-a，也就是说，泛化关系可以理解为典型的父类-子类关系，其次还可以理解为A有n种B的情况下，前者是后者的泛化也就是父类。泛化关系中，父类是抽象用例，抽象后，必须且只能选择一种子类用例去执行。

对比来看，用例之间的关系（用例之间存在3种关系--包含、扩展、泛化）

包含关系<<include>>：将用例中的一部分行为抽取出来作为单独的用例，对于被包含的用例，是必须选择的用例；

扩展关系<<extend>>：将某个条件下可能的行为作为扩展用例，是某个条件下可以选择的扩展；

泛化关系：将用例中的一部分公共部分抽象出来作为父用例，对于这些子用例，必须选择其中一个用例。（注意与包含关系的区分）

3、对于面向对象程序设计语言，选择哪一种？掌握基础语法应该掌握到什么程度？

答：

面向对象程序设计题是软设考试下午题的试题五、试题六，分别使用的是C++和JAVA，可以二选一。建议大家选择自己熟悉的语言。如果二者都不熟悉，建议选择JAVA进行学习。

掌握基础语法，要求能够读懂程序代码，并了解程序的结构和调用关系等。

# 第7章 数据结构

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解数组与矩阵按行存储、按列存储的区别，掌握相对位置的计算方式。

（2）了解顺序表与链表的区别，以及不同维度的对比（空间性能、时间性能）。

（3）熟悉特殊的线性结构——栈与队列的序列问题（队列——先进先出；栈——先进后出），了解字符串的特点和操作。

（4）了解广义表的一些概念和操作（长度、深度、取头、取尾等）。

（5）了解树与二叉树的一些基本概念，了解二叉树的一些特性。

（6）掌握二叉树的遍历（前序遍历-根-左-右、后序遍历-左-右-根、中序遍历-左-根-右、层次遍历）及其反向构造过程。

（7）了解特殊的二叉树（查找二叉树、最优二叉树、平衡二叉树）。

（8）了解图的基本概念，了解图的存储方式（邻接矩阵、邻接表），了解图的遍历（深度遍历、广度遍历），了解图的拓扑排序，了解最小生成树。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 数组与矩阵（★★） | 数组与矩阵 |
| 1 | 线性结构（★★★★★） | 线性结构 |
| 1 | 广义表（★） | 广义表 |
| 1 | 树与二叉树（★★★★★） | 树与二叉树 |
| 1 | 图（★★） | 图 |

## 2 知识点详情

### 2.1 数组与矩阵（★★）

【考法分析】

1、本知识点的考查形式主要有：给定一些数组或矩阵，计算对应某个元素的存放位置或位置的表示公式。

【要点分析】

1、对于数组或矩阵，存储时注意存储方式是按行存储还是按列存储，二者结果有区别。

2、对于**存储位置的计算**，可以理解为计算当前位置以要求的存储方式存放时，前面已经存放了多少个元素。

|  |  |
| --- | --- |
| 数组类型 | 存储地址计算 |
| 一维数组a[n] | a[i]的存储地址为：a+i\*len |
| 二维数组a[m][n] | a[i][j]的存储地址（按行存储）为：a+(i\*n+j)\*len  a[i][j]的存储地址（按列存储）为：a+(j\*m+i)\*len |



【备考点拨】

1、对于某些相对繁杂的数组或矩阵，建议可以将前几个特殊的元素（如a[0][0]等）代入验证公式，排除错误的选项，直到找出正确选项。

### 2.2 线性结构（★★★★★）

【考法分析】

1、本知识点的主要考查形式有：对顺序表和链表的一些特点描述判断正误；或对顺序表和链表的一些操作进行对比；对于特殊的线性表队列和栈的一些概念描述判断正误，或二者的出入序列合法性的判断。

【要点分析】

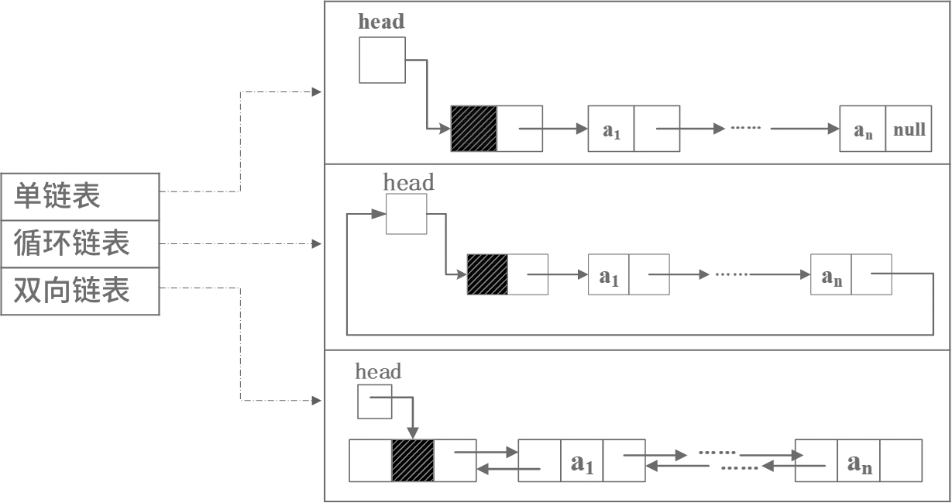
1、顺序表和链表的对比：

|  |  |  |  |
| --- | --- | --- | --- |
| 性能类别 | 具体项目 | 顺序存储 | 链式存储 |
| 空间性能 | 存储密度 | =1，更优 | <1 |
| 容量分配 | 事先确定 | 动态改变，更优 |
| 时间性能 | 查找运算 | O(n) | O(n) |
| 读运算 | O(1)，更优 | O(n)，最好情况为1，最坏情况为n |
| 插入运算 | O(n)，最好情况为0，最坏情况为n | O(1)，更优 |
| 删除运算 | O(n) | O(1)，更优 |

2、**顺序表**：线性表顺序存储，即用一组地址连续的存储单元依次存储线性表中的数据元素，从而使得逻辑上相邻的两个元素，在物理上也相邻。在存储之前，先根据线性表的长度分配连续的物理空间，因此后续不方便扩展。只需要存储数据元素，不需要存储元素的逻辑关系因此存储密度为1。

3、**链表**：线性表链式存储，即用通过指针链接起来的结点来存储数据元素，存储各数据元素的结点物理上不要求连续，因此后期扩展方便。因为物理上不连续，需要同时存储各元素之间的逻辑关系，存储密度小于1。

4、链表的分类：单链表、双链表、循环链表。



5、特殊的线性结构：队列（先进先出）、栈（先进后出）。

6、特殊的线性结构：字符串

串的定义：串是仅由字符构成的有限序列，是一种线性表。一般记为S=“a1a2a3…an”，其中，S是串名，单引号括起来的字符序列是串值。

基本概念

（1）空串与空格串

空串：长度为零，不包含任何字符。

空格串：由一个或多个空格组成的串。虽然空格是一个空白字符，但它也是一个字符，在计算串长度时要将其计算在内。

（2）子串与子序列

子串：由串中任意长度的连续字符构成的序列称为子串。含有子串的串称为主串。子串在主串中的位置是指子串首次出现时，该子串的第一个字符在主串中的位置。空串是任意串的子串。

子序列：一个串的“子序列”（subsequence）是将这个串中的一些字符提取出来得到一个新串，并且不改变它们的相对位置关系。

（3）串比较与串相等

串比较：两个串比较大小时以字符的ASCII码值（或其他字符编码集合）作为依据。实质上，比较操作从两个的第一个字符开始进行，字符的码值大者所在的串为大；若其中一个串先结束，则以串长较大者为大。

串相等：指两个串长度相等且对应序号的字符也相同。

串的基本操作：

（1）赋值操作StrAssign(s,t)：将串s的值赋给串t。

（2）连接操作Concat(s,t)：将串t接续在串s的尾部，形成一个新的串。

（3）求串长StrLength(s)：返回串s的长度。

（4）串比较StrCompare(s,t)：比较两个串的大小。返回值-1、0和1分别表示s<t、s=t和s>t三种情况。

（5）求子串SubString(s,start,len)：返回串S中从start开始的、长度为len的字符序列。

**模式匹配**：子串的定位操作通常称为串的模式匹配。（子串也称为模式串）

**KMP算法**：其改进之处在于——每当匹配过程中出现相比较的字符不相等时，不需要回退到主串的字符位置指针，而是利用已经得到的“部分匹配”结果将模式串向右“滑动”尽可能远的距离，再继续进行比较。在KMP算法中，依据模式串的next函数值实现子串的滑动。若令next[j]=k，则next[j]表示当模式串中的pj与主串中相应字符不相等时，令模式串的pnext[j]与主串的相应字符进行比较。(**j=next[j]**)

7、循环队列：

队空条件：head=tail

队满条件：(tail+1)%size=head

【备考点拨】

1、掌握顺序表和链表各自的特点，能够加以区分，并判断相关描述的正确性；

2、了解顺序表和链表一些操作的特殊性和对比；

3、对于队列和栈，掌握相关的特点和一些特殊的操作、循环队列相关判断公式；

4、掌握队列的入队和出队序列的特点；掌握栈的入栈和出栈序列的特点。

### 2.3 广义表（★★）

【考法分析】

1、对于本知识点的主要考查形式有：对相关概念的描述判断正误；给定广义表，指出得到对应结果所需的运算过程。

【要点分析】

1、广义表是n个表元素组成的有限序列，是线性表的推广。

2、通常用递归的形式进行定义，记作：LS=(a0, a1,…, an)。

注：其中LS是表名，ai是表元素，它可以是表（称作子表），也可以是数据元素（称为原子）。其中n是广义表的长度（也就是最外层包含的元素个数），n=0的广义表为空表；而递归定义的重数就是广义表的深度，直观地说，就是定义中所含括号的重数（原子的深度为0，空表的深度为1）。

3、基本运算：取表头head(Ls)和取表尾tail(Ls)。

**取表头head(Ls)**，非空广义表的Ls的第一个元素称为表头，它可以是一个单元素，也可以是一个子表。

**取表尾tail(Ls)**，非空广义表Ls，除表头元素之外，由其余元素所构成的表称为表尾。非空广义表的表尾必定是一个表。

若有：LS1=（a，（b，c），（d，e））

head(LS1)＝a

tail(LS1)=(（b，c），（d，e）)

【备考点拨】

1、了解广义表相关的一些概念；

2、掌握广义表的相关运算。

### 2.4 树与二叉树（★★★★★）

【考法分析】

1、本知识点的主要考查形式有：对数与二叉树的一些概念和特性的描述，判断其正误；对于特殊的二叉树（平衡树、哈弗曼树、满二叉树、排序树等）定义、特性的描述判断正误、或根据题干描述构造特殊的二叉树，找到对应的选项；考查二叉树的遍历结果，或根据遍历序列，找到对应的二叉树。

【要点分析】

1、树与二叉树的特性：

（1）树的概念：

**双亲、孩子和兄弟**：结点的子树的根称为该结点的孩子；相应地，该结点称为其子结点的双亲。具有相同双亲的结点互为兄弟。

（这里涉及到2个层次，第一个层次的子树，这棵子树的根是第一层结点的孩子结点，第一层结点是其子节点的双亲节点/父节点）。

**结点的度**：一个结点的子树的个数记为该结点的度

**叶子节点**：也称为终端结点，指度为0的结点

**内部结点**：指度不为0的结点，也称为分支节点或非终端节点。除根结点之外，分支结点也称为内部结点。

**结点的层次**：根为第一层，根的孩子为第二层，依次类推，若某节点在第i层，则其孩子结点在第i+1层

**树的高度**：一颗树的最大层次数记为树的高度（深度）

（2）二叉树的重要特性：

1、在二叉树的第i层上最多有2i-1个结点（i≥1）；

2、深度为k的二叉树最多有2k -1个结点（k≥1）；

3、对任何一棵二叉树，如果其叶子结点数为n0，度为2的结点数为n2，则**n0=n2+1**。

4、如果对一棵有n个结点的完全二叉树的结点按层序编号（从第1层到层，每层从左到右），则对任一结点i（1≤i≤n），有：

如果i=1，则结点i无父结点，是二叉树的根；如果i>1，则父结点是⎣i/2⎦ ；

如果2i>n，则结点i为叶子结点，无左子结点；否则，其左子结点是结点2i；

如果2i+1>n，则结点i无右子叶点，否则，其右子结点是结点2i+1。

2、特殊的树

**二叉树**：二叉树是每个结点最多有两个孩子的有序数，可以为空树，可以只有一个结点。

**满二叉树**：任何结点，或者是树叶，或者恰有两棵非空子树。

**完全二叉树**：最多只有最小面的两层结点的度可以小于2，并且最下面一层的结点全都集中在该层左侧的若干位置。

**平衡二叉树**：树中任一结点的左右子树高度之差不超过1。

**查找二叉树**：又称之为排序二叉树。任一结点的权值，大于其左孩子结点，小于其右孩子结点。

**线索二叉树**：在每个结点中增加两个指针域来存放遍历时得到的前驱和后继信息。

**最优二叉树**：又称为哈弗曼树，它是一类带权路径长度最短的树。

**路径**是从树中一个结点到另一个结点之间的通路，路径上的分支数目称为路径长度。

**树的路径**长度是从树根到每一个叶子之间的路径长度之和。结点的带权路径长度为从该结点到树根之间的路径长度与该结点权值的乘积。

**树的带权路径长度**为树中所有叶子结点的带权路径长度之和。

**哈弗曼树的构造**：（1）根据给定的权值集合，找出最小的两个权值，构造一棵子树将这两个权值作为其孩子结点，二者权值之和作为根结点；（2）在原集合中删除这两个结点的权值，并引入根节点的权值；（3）重复步骤（1）和步骤（2），直到原权值集合为空。——自己操作熟悉

**哈夫曼编码**：根据哈夫曼树进行边长编码，编码长度与路径长度相关，左侧分支编码为0（或1），右侧分支编码为1（或0），从根结点到对应叶子结点所有路径分支上的编码记录下来，即为该叶子结点的编码。

3、树的遍历操作：遍历是按某种策略访问树中的每个结点，且仅访问一次的过程。

前序遍历：又称为先序遍历，按根🡪左🡪右的顺序进行遍历。

后序遍历：按左🡪右🡪根的顺序进行遍历。

中序遍历：按左🡪根🡪右的顺序进行遍历。

层次遍历：按层次顺序进行遍历。

【备考点拨】

1、掌握树相关的概念和特性；

2、掌握一些特殊的树的定义和特性；

3、掌握哈夫曼树的构造过程，哈夫曼编码的构造。

4、掌握树的遍历，能够根据树的遍历序列反向构造二叉树的过程。

### 2.5 图（★★）

【考法分析】

1、本知识点的主要考查形式有：判断给出的关于图的概念、特性的描述是否正确；或根据图的邻接矩阵、邻接表，指出相关图、图的特点、图的遍历；根据图示，指出遍历顺序、拓扑序列。

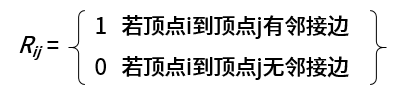
【要点分析】

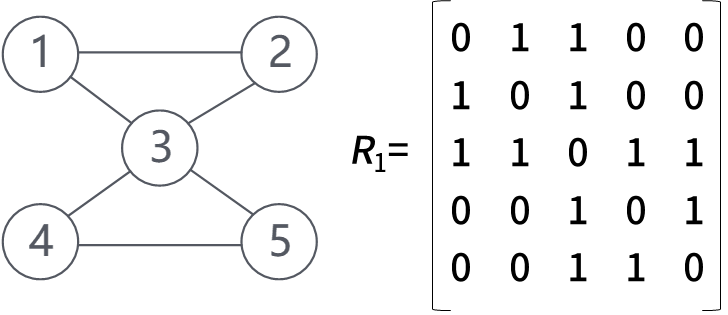
1、完全图

在无向图中，若每对顶点之间都有一条边相连，则称该图为完全图（complete graph）。

在有向图中，若每对顶点之间都有二条有向边相互连接，则称该图为完全图。

2、图的邻接矩阵表示：用一个n阶方阵R来存放图中各结点的关联信息，其矩阵元素Rij定义为：

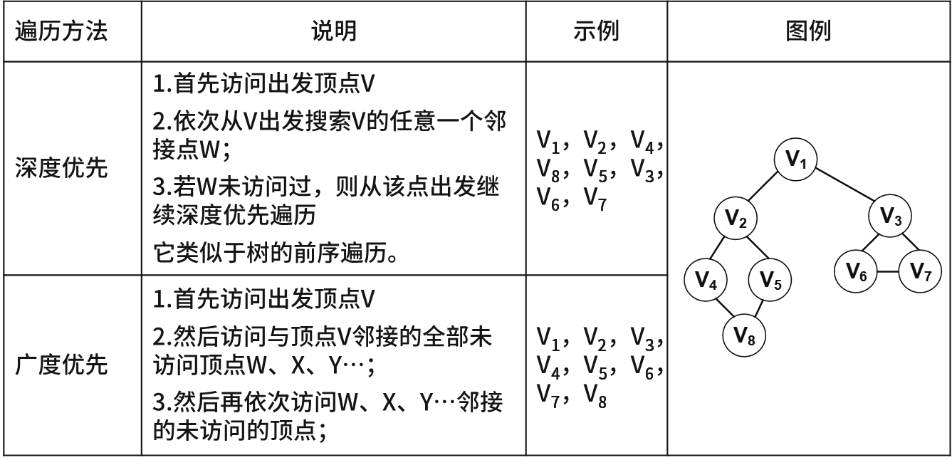




3、图的邻接表表示：首先把每个顶点的邻接顶点用链表示出来，然后用一个一维数组来顺序存储上面每个链表的头指针。



4、图的遍历：



5、图的拓扑排序：拓扑排序是将AOV网中的所有顶点排成一个线性序列的过程，并且该序列满足：若在AOV网点中从顶点Vi到Vj有一条路径，则在该线性序列中，顶点Vi必然在顶点Vj之前。

【备考点拨】

1、掌握图的相关概念；

2、掌握图的存储；

3、掌握图的遍历；

4、掌握图的拓扑序列求取。

## 3 章节问答

1、数据结构会不会出现在算法题中？

答：

数据结构是数据的组织形式，算法与数据结构有紧密的关系。数据结构不同，对算法的时间、空间效率都有可能产生影响。数据结构与算法的结合应用，是必定存在，结合的可能是简单的数组结构，也可能是复杂的链表结构等。掌握相应数据结构的基本操作，结合语法能够读懂和调用即可。

2、特殊的矩阵如何记住相应特点以推导出计算公式？

答：

这一部分不建议按规律推导，对于特殊的二维矩阵没有具体说明时，在不同的教材中也会存在定义的细微差别，对于本类试题，建议以图示已经出现的2-3个元素直接进行验证选项即可。

3、二叉树的特性记不住怎么办？

答：

对于树与二叉树的特性描述较多，这一类不建议直接背诵，可以画出一棵实际的树来进行验证或者找反例。比较特殊的**n0=n2+1，**涉及到二叉树中经常会用到的一个分析思想：度为0的结点记作n0，度为1的结点记作n1，度为2的结点记作n2。首先从根出发，向叶子结点进行分析，度定义的是分支的数量，分支总数为0\*n0+1\*n1+2\*n2；从叶子结点向根反向分析，每个结点都会通过分支指向其父结点，除了根结点，反向分析的分值总数为n0+n1+n2-1。二者相等，调整后结果即为**n0=n2+1**。

4、图的最小生成树需要掌握到什么程度？

答：

对于图的最小生成树，目前在软设中的考查形式只是了解相应的算法用到贪心策略即可，不需要了解详细的分析过程，如果在算法题中涉及，也会用自然语言先做描述介绍，不需要考生记忆。

# 第8章 算法基础

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解算法的特性，了解常用算法的特征（分治法、动态规划法、回溯法、贪心法）。

（2）了解时间复杂度的量级，大小排序，以及应用情景，能够判断出常见过程的时间复杂度。

（3）了解顺序查找过程，熟悉二分查找的过程，了解散列表的构造和冲突的解决。复杂度，了解各种排序算法的排序过程。

（4）掌握排序算法的分类和相关算法的特点对比，了解各种排序算法的时间复杂度和空间。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 算法基础与常见算法（★★★★★） | 算法基础与常见算法 |
| 1 | 时间复杂度与空间复杂度（★★★★★） | 时间复杂度与空间复杂度 |
| 1 | 查找（★★★★★） | 查找 |
| 1 | 排序（★★★★★） | 排序 |

## 2 知识点详情

### 2.1 算法基础及常见算法（★★★★★）

【考法分析】

1、本知识点的考查形式主要有：根据题干的情景描述，判断所使用的算法策略；判断算法相关描述是否正确；下午题也会考查根据题干说明和代码，判断算法策略。

【要点分析】

1、算法的特性：

（1）**有穷性**：执行有穷步之后结束，且每一步都可在有穷时间内完成。

（2）**确定性**：算法中每一条指令都必须有确切的含义，不能含糊不清。

（3）输入（>=0）

（3）输出（>=1）

（4）**有效性（可行性）**：算法的每个步骤都能有效执行并能得到确定的结果。例如a=0，b/a就无效

2、分治法

（1）特征：把一个问题拆分成多个小规模的相同子问题，一般可用递归解决。

（2）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、大整数乘法、汉诺塔。

3、动态规划法（用于求最优解）——“最优子结构”和递归式

（1）特征：划分子问题（最优子结构），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。

（2）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

4、回溯法

（1）特征：系统的搜索一个问题的所有解或任一解。有试探和回退的过程。

（2）经典问题：N皇后问题、迷宫、背包问题。

5、**贪心法**（一般用于求满意解）

（1）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。

（2）经典问题：背包问题（如装箱）、多机调度、找零钱问题。

6、常见算法策略对比

|  |  |  |  |
| --- | --- | --- | --- |
| 算法名称 | 关键点 | 特征 | 典型问题 |
| 分治法 | 递归技术 | 把一个问题拆分成多个小规模的相同子问题，一般可用递归解决。 | 归并排序、快速排序、二分搜索 |
| 贪心法 | 一般用于求满意解，特殊情况可求最优解（部分背包） | 局部最优，但整体不见得最优。每步有明确的，既定的策略。 | 背包问题（如装箱）、多机调度、找零钱问题 |
| 动态规划法 | 最优子结构和递归式 | 划分子问题（最优子结构），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。 | 矩阵乘法、背包问题、 LCS最长公共子序列 |
| 回溯法 | 探索和回退 | 系统的搜索一个问题的所有解或任一解。有试探和回退的过程。 | N皇后问题、迷宫、背包问题 |

【备考点拨】

1、掌握算法的特性、概念；

2、掌握常见算法的特点、适用场景，并能够加以区分。

### 2.2 时间复杂度与空间复杂度（★★★★★）

【考法分析】

1、本知识点的考查形式主要有：根据题干描述的情景，根据排序方法、算法逻辑或相关代码，计算其时间复杂度或空间复杂度；根据递归式，计算其时间复杂度；下午题也会考查根据题干说明和代码，指出时间复杂度。

【要点分析】

1、时间复杂度是指程序运行从开始到结束所需要的时间。通常分析时间复杂度的方法是从算法中选取一种对于所研究的问题来说是基本运算的操作，以该操作重复执行的次数作为算法的时间度量。一般来说，算法中原操作重复执行的次数是规模n的某个函数T(n)。由于许多情况下要精确计算T(n)是困难的，因此引入了渐进时间复杂度在数量上估计一个算法的执行时间。其定义如下：

如果存在两个常数c和m，对于所有的n，当n≥m时有f(n)≤cg(n)，则有f(n)=O(g(n))。也就是说，随着n的增大，f(n)渐进地不大于g(n)。例如，一个程序的实际执行时间为T(n)=3n3+2n2+n，则T(n)=O(n3)。

常见的对算法执行所需时间的度量：

O(1)<O(log2n)<O(n)<O(nlog2n)<O(n2)<O(n3)<O(2n)

2、常见算法逻辑的时间复杂度：

（1）单个语句，或程序无循环和复杂函数调用：O(1)

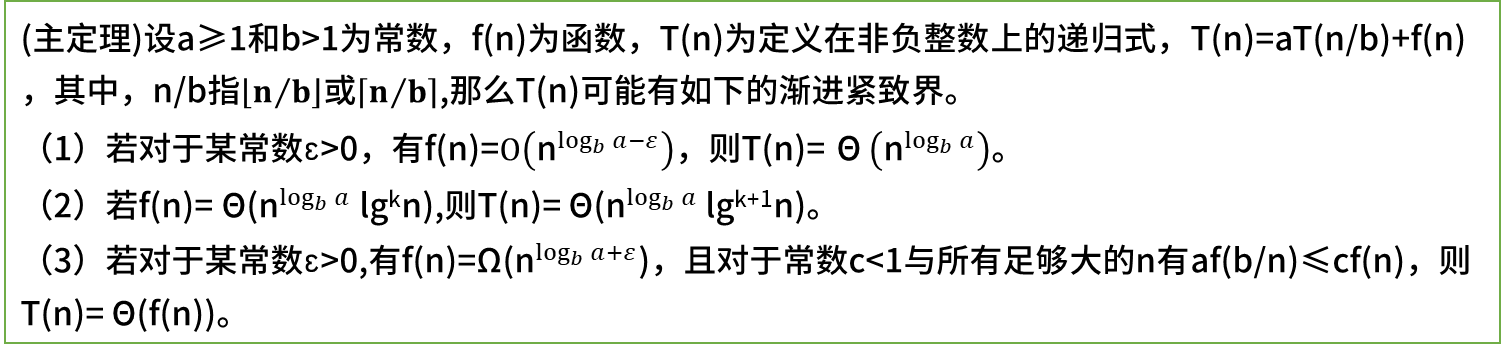
（2）单层循环：O(n)；双层嵌套循环：O(n2)；三层嵌套循环：O(n3)。

（3）树形结构、二分法、构建堆过程：O(log2n)。

（4）堆排序、归并排序：O(nlog2n)。

（5）所有不同可能的排列组合：O(2n)。

3、主定理求固定形式递归式的时间复杂度：



【备考点拨】

1、掌握常见排序算法的时间复杂度和空间复杂度；

2、掌握常见排序算法、常见算法逻辑（如循环）的时间复杂度；

3、了解主定理求取递归式的时间复杂度。

### 2.3查找（★★★★★）

【考法分析】

1、本知识点的主要考查形式有：指出二分查找的比较次数、比较对象、时间复杂度；指出散列表查找的相关概念描述是否正确。

【要点分析】

1、顺序查找的思想：将待查找的关键字为key的元素从头到尾与表中元素进行比较，如果中间存在关键字为key的元素，则返回成功；否则，则查找失败。

2、二分法查找的基本思想是：（设R[low,…,high]是当前的查找区）

（1）确定该区间的中点位置：mid=L(low+high)/2˩；

（2）将待查的k值与R[mid].key比较，若相等，则查找成功并返回此位置，否则需确定新的查找区间，继续二分查找，具体方法如下。

若R[mid].key＞k，则由表的有序性可知R[mid,…,n].key均大于k，因此若表中存在关键字等于k的结点，则该结点必定是在位置mid左边的子表R[low,…,mid–1]中。因此，新的查找区间是左子表R[low,…,high]，其中high=mid–1。

若R[mid].key<k，则要查找的k必在mid的右子表R[mid+1,…,high]中，即新的查找区间是右子表R[low,…,high]，其中low=mid+1。

若R[mid].key=k，则查找成功，算法结束。

（3）下一次查找是针对新的查找区间进行，重复步骤（1）和（2）。

（4）在查找过程中，low逐步增加，而high逐步减少。如果high<low，则查找失败，算法结束。

折半查找在查找成功时关键字的比较次数最多为次。

折半查找的时间复杂度为 O(log2n)次。

【例题】请给出在含有12个元素的有序表｛1，4，10，16，17，18，23，29，33，40，50，51｝中二分查找关键字17的过程。



比较次数：4次；比较对象a[6]，a[3]，a[4]，a[5]。

3、散列表查找的基本思想是：已知关键字集合U，最大关键字为m，设计一个函数Hash，它以关键字为自变量，关键字的存储地址为因变量，将关键字映射到一个有限的、地址连续的区间T[0..n-1](n<<m)中，这个区间就称为散列表，散列查找中使用的转换函数称为散列函数。

开放定址法是指当构造散列表发生冲突时，使用某种探测手段，产生一个探测的散列地址序列，并且逐个查找此地址中是否存储了数据元素，如果没有，则称该散列地址开放，并将关键字存入，否则冲突，继续查找下一个地址。

例：记录关键码为（3，8，12，17，9），取m=10（存储空间为10），p=5，散列函数h=key%p。



### 2.4 排序（★★★★★）

【考法分析】

1、本知识点的主要考查形式有：给定情景描述，指出适用的排序方法；指出特定排序方法的时间复杂度、空间复杂度。

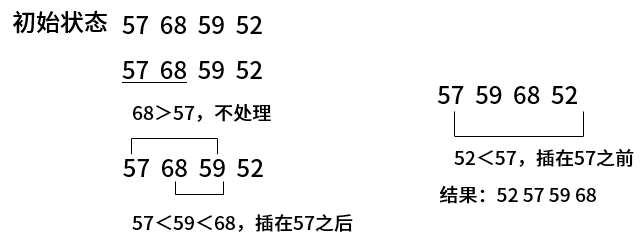
【要点分析】

1、排序分类

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 类别 | 排序方法 | 时间复杂度 | | 空间复杂度 | 稳 定 性 |
| 平均情况 | 特殊情况 | 辅助存储 |
| 插入排序 | 直接插入 | O(n2) | 基本有序最优O(n) | O(1) | 稳定 |
| Shell排序 | O(n1.3) | - | O(1) | 不稳定 |
| 选择排序 | 直接选择 | O(n2) | - | O(1) | 不稳定 |
| 堆排序 | O(nlog2n) | - | O(1) | 不稳定 |
| 交换排序 | 冒泡排序 | O(n2) | - | O(1) | 稳定 |
| 快速排序 | O(nlog2n) | 基本有序最差(n2) | O(log2n) | 不稳定 |
| 归并排序 | | O(nlog2n) | -- | O(n) | 稳定 |
| 基数排序 | | O(d(n+rd)) | -- | O(rd) | 稳定 |

2、直接插入排序：即当插入第i个记录时，R1，R2，…，Ri-1均已排好序，因此，将第i个记录Ri依次与Ri-1，…，R2，R1进行比较，找到合适的位置插入。它简单明了，但速度很慢。

注：对于基本有序的序列，选择直接插入排序方法，时间复杂度近乎线性为：O(n)。



3、希尔（Shell）排序：先取一个小于n的整数d1作为第一个增量，把文件的全部记录分成d1个组。所有距离为d1的倍数的记录放在同一个组中。先在各组内进行直接插入排序；然后，取第二个增量d2<d1重复上述的分组和排序，直至所取的增量dt=1(dt<dt–1<O<d2<d1)，即所有记录放在同一组中进行直接插入排序为止。该方法实质上是一种分组插入方法。



4、直接选择排序的过程是，首先在所有记录中选出排序码最小的记录，把它与第1个记录交换，然后在其余的记录内选出排序码最小的记录，与第2个记录交换……依次类推，直到所有记录排完为止。



5、堆排序

（1）堆的定义：设有n个元素的序列{K1，K2，…，Kn}，当且仅当满足下述关系之一时，称之为堆。

小顶堆：Ki≤K2i 且Ki≤K2i +1；

大顶堆：Ki≥K2i 且Ki≥K2i +1 。

（2）堆排序的基本思想为：先将序列建立堆，然后输出堆顶元素，再将剩下的序列建立堆，然后再输出堆顶元素，依此类推，直到所有元素均输出为止，此时元素输出的序列就是一个有序序列。

（3）堆排序的算法步骤如下（以大顶堆为例）：

初始时将顺序表R[1…n]中元素建立为一个大顶堆，堆顶位于R[1]，待序区为R[1…n]。

循环执行步骤3～步骤4，共n-1次。

假设为第i 运行，则待序区为R[1…n-i+1]，将堆顶元素R[1]与待序区尾元素R[n-i+1]交换，此时顶点元素被输出，新的待序区为R[1…n-i ]。

待序区对应的堆已经被破坏，将之重新调整为大顶堆。

（4）堆排序时间复杂度为：O(nlog2n)，是不稳定的排序。

6、冒泡排序的基本思想是，通过相邻元素之间的比较和交换，将排序码较小的元素逐渐从底部移向顶部。由于整个排序的过程就像水底下的气泡一样逐渐向上冒，因此称为冒泡算法。



7、快速排序采用的是分治法，其基本思想是将原问题分解成若干个规模更小但结构与原问题相似的子问题。通过递归地解决这些子问题，然后再将这些子问题的解组合成原问题的解。

快速排序通常包括两个步骤：

第一步，在待排序的n个记录中任取一个记录，以该记录的排序码为准，将所有记录都分成两组，第1组都小于该数，第2组都大于该数，如图所示。

第二步，采用相同的方法对左、右两组分别进行排序，直到所有记录都排到相应的位置为止。



8、归并也称为合并，是将两个或两个以上的有序子表合并成一个新的有序表。若将两个有序表合并成一个有序表，则称为二路合并。合并的过程是：比较A[i]和A[j]的排序码大小，若A[i]的排序码小于等于A[j]的排序码，则将第一个有序表中的元素A[i]复制到R[k]中，并令i和k分别加1；如此循环下去，直到其中一个有序表的元素复制完，然后再将另一个有序表的剩余元素复制到R中。



9、基数排序是一种借助多关键字排序思想对单逻辑关键字进行排序的方法。基数排序不是基于关键字比较的排序方法，它适合于元素很多而关键字较少的序列。基数的选择和关键字的分解是根据关键字的类型来决定的，例如关键字是十进制数，则按个位、十位来分解。



10、各种排序算法应用情景对比：

若待排序列的记录数目n较小，可采用直接插入排序和简单选择排序。由于直接插入排序所需的记录移动操作较简单选择排序多，因而当记录本身信息量大时，用简单选择排序方法较好。

若待排记录按关键字基本有序，宜采用直接插入排序或冒泡排序。

当n很大且关键字位数较少时，采用基数排序较好。

若n很大，则应采用时间复杂度为O(nlog2n)的排序方法，例如快速排序、堆排序或归并排序：

（1）快速排序目前被认为是内部排序中最好的方法，当待排序的关键字为随机分布时，快速排序的平均运行时间最短；

（2）堆排序只需要一个辅助空间，并且不会出现在快速排序中可能出现的最快情况。

（3）快速排序和堆排序都是不稳定的排序方法，若要求排序稳定，可选择归并排序。

【备考点拨】

1、掌握顺序查找的相关概念；

2、掌握二分查找的过程；

3、掌握散列表的位置分布和冲突相关的概念；

4、掌握常见排序方法的分类、时间复杂度、空间复杂度、稳定性等；

5、掌握常见排序方法的排序过程（堆排序了解其排序过程）。

# 第9章 程序设计语言与语言处理程序基础

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解编译与解释，区分二者过程和特点；

（2）了解文法概念和分类（不要求掌握），了解不同类型文法与自动机的对应关系；

（3）了解文法推导树（根据字符串画树）；

（4）了解有限自动机的概念；

（5）了解正规式的概念，熟悉常见的正规集表示，掌握有限自动机与正规式的结合考查题型；

（6）了解数据类型和程序控制结构；

（7）掌握后缀表达式；

（8）掌握函数调用-传值与传址的应用和区别；

（9）了解常见程序语言的特点。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 编译与解释（★★★） | 编译与解释 |
| 1 | 文法（★★） | 文法 |
| 1 | 正规式（★★★★） | 正规式 |
| 1 | 有限自动机（★★） | 有限自动机 |
| 1 | 后缀表达式（★★★） | 后缀表达式 |
| 1 | 传值与传址（★★★★） | 传值与传址 |
| 1 | 多种程序设计语言特点（★★★） | 多种程序设计语言特点 |

## 2 知识点详情

### 2.1 编译与解释（★★★）

【考法分析】

1、本知识点的考查形式主要有：给出编译与解释相关的描述，判断正误；给出编译各个阶段的描述，判断正误。

【要点分析】

1、**解释程序**，也称解释器；直接解释执行源程序，或者将源程序翻译成某种中间代码后再加以执行。

2、**编译程序**，也称编译器；将源程序翻译成目标语言程序，然后在计算机上运行目标程序。

3、**两者的根本区别**：编译方式下，机器上运行的是与源程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程，因此执行时效率较高；解释方式下，解释程序和源程序（或某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序，边解释边执行，执行效率较低。即：解释方式，翻译程序不生成独立的目标程序，而编译方式则生成独立保持的目标程序。

|  |  |  |  |
| --- | --- | --- | --- |
|  | | 编译型语言 | 解释型语言 |
| 共同点 | | 高级程序语言 | |
| 有词法分析、语法分析、语义分析过程 | |
| 不同点 | 翻译程序 | 编译器 | 解释器 |
| 是否生成目标代码 | 生成目标代码 | 不会生成目标代码 |
| 目标程序能够直接执行 | 目标程序直接执行 | 边解释边执行 |
| 翻译程序是否参与执行 | 编译器不参与执行 | 解释器参与执行 |
| 执行效率 | 执行效率高 | 执行效率低 |
| 灵活性与可移植性 | 灵活性差，可移植性差 | 灵活性好，可移植性强 |

4、编译过程：

（1）**词法分析阶段**：是编译过程的第一阶段，其任务是对源程序从前到后（从左到右）逐个字符扫描，从中识别出一个个“单词”符号。词法分析过程的依据是语言的词法规则，即描述“单词”结构的规则。

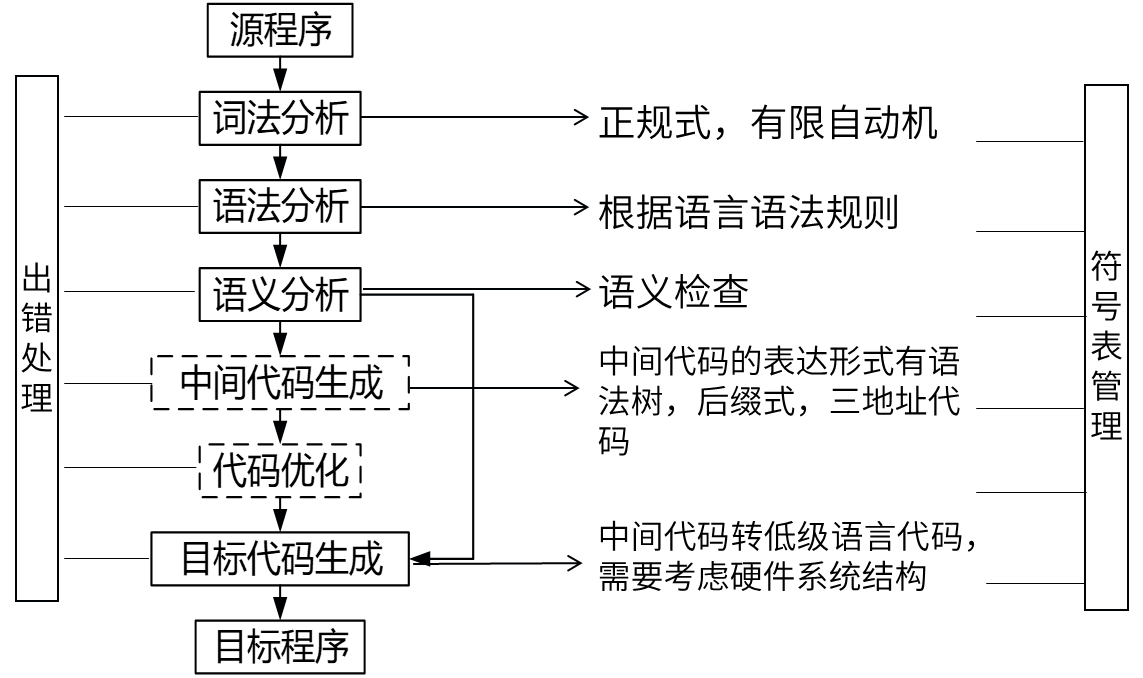
（2）**语法分析阶段**：其任务是在词法分析的基础上，根据语言的语法规则将单词符号序列分解成各类语法单位。通常语法分析是确定整个输入串是否构成一个语法上正确的程序。一般来说，通过编译的程序，不存在语法上的错误。

（3）**语义分析阶段**：其任务主要检查源程序是否包含静态语义错误（动态语义错误在执行过程中才能发现），并收集类型信息供后面的代码生成阶段使用。语义分析的一个主要工作是进行类型分析和检查。

（4）**中间代码生成**：其任务是根据语义分析的输出生成中间代码。**此阶段不是必须的**。常见的中间代码有：树、后缀式、三地址码（四元式）。

（5）**代码优化**：其任务是优化中间代码。**此阶段不是必须的**。

（6）**目标代码生成**：是编译器工作的最后一个阶段。其任务是把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码或汇编指令代码。本阶段与具体机器密切相关。



（7）**符号表**的作用是记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。符号表的存在可以贯穿编译所有阶段。

（8）错误管理

静态错误：编译时所发现的程序错误，分为语法错误和静态语义错误。

**语法错误包含**：单词拼写错误、标点符号错误、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。

**静态语义分析**：运算符与运算对象类型不合法等错误。

**动态错误**：发生程序运行时，也叫动态语义错误。包括死循环、变量取零时做除数、引用数组元素下标越界等错误。

【备考点拨】

1、掌握编译与解释的区别；

2、掌握编译器的工作过程。

3、了解错误的分类以及常见错误。

### 2.2 文法（★★）

【考法分析】

1、本知识点的主要考查方式有：给出一些概念的描述判断正误；给出一个文法的描述，判断能够识别的字符串。

【要点分析】

1、文法相关的概念：一个形式文法是一个有序四元组G=(V，T，S，P)，其中：

V：**非终结符**。不是语言组成部分，不是最终结果，可理解为占位符。

T：**终结符**。是语言的组成部分，是最终结果。 V∩T=∅

S：**起始符**。是语言的开始符号。

P：**产生式**。用终结符替代非终结符的规则。形如α→β

2、文法的分类：

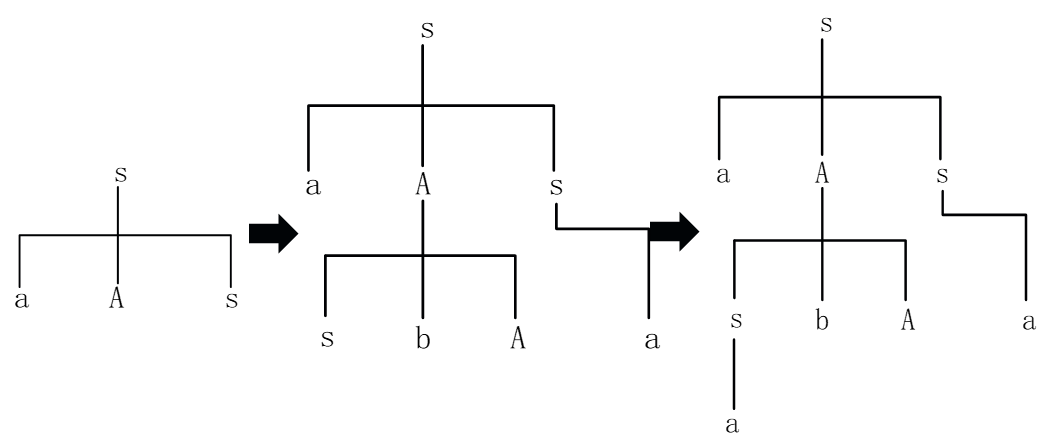
|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 别称 | 说明 | 对应自动机 |
| 0型 | 短语文法 | G的每条产生式α→β满足a属于V’的正则闭包，且至少含有一个非终结符，而β属于V’的闭包 | 图灵机 |
| 1型 | 上下文有关文法 | G的任何产生式α→β满足|α|<=|β|，仅仅S→ε例外，但S不得出现在任何产生式右部 | 线性界限自动机 |
| 2型 | 上下文无关文法 | G的任何产生式为A→β，A为非终结符，β为V’的闭包 | 非确定的下推自动机 |
| 3型 | 正规文法 | G的任何产生式为A→αB或A→α，α属于非终结符的闭包，A、B都属于非终结符 | 有限自动机 |

注：常见的程序设计语言一般是上下文无关文法。

3、文法与语法树的推导：

例：文法G=（{a, b}, {S, A}, S, P），其中：S→aAS|a；A→SbA|SS|ba。请构造句型aabAa的推导树。

S → aAS； S → a； A → SbA；A → SS； A → ba。



【备考点拨】

1、掌握文法相关的概念和分类、对应的自动机；

2、掌握语法推导树。

### 2.3 正规式（★★★★）

【考法分析】

1、本知识点的主要考查形式有：给出正规式指出符合要求的字符串；或给出一些相关的描述判断其正误；或结合有限自动机考查。

【要点分析】

1、正规式是描述程序语言单词的表达式，对于字母∑，其上的正规式及其表示的正规集可以递归定义如下。

① ε是一个正规式，它表示集合L(ε)={ε}。

② 若a是∑上的字符，则a是一个正则式，它所表示的正规L(a)={a}。

③ 若正规式r和s分别表示正规集L(r)=L(s)，则

（a）r|s是正规式，表示集合L(r)∪L(s)；

（b）r·s是正规式，表示集合L(r)L(s)；

（c）r\*是正规式，表示集合(L(r))\*；

（d）(r)是正规式，表示集合L(r)。

仅由有限次地使用上述三个步骤定义的表达式才是∑上的正规式。由此可见，正规式要么为空，要么由字母、或、连接、闭包运算符组成。其中闭包运算符“\*”具有最高的优先级，连接运算具有次高优先级，或运算符“|”具有最低优先级。

2、常见正规式的含义：

|  |  |  |
| --- | --- | --- |
| 正规式 | 正规集 | 举例 |
| ab | 字符串ab构成的集合 | {ab} |
| a|b | 字符串a、b构成的集合 | {a , b } |
| a\* | 由0或多个a构成的字符串集合 | {空 , a , aa , aaa , a…a(n个a )} |
| (a|b)\* | 所有字符a和b构成的串的集合 | {空 , a , b , ab , aab , abb , baa, aba , …} |
| a(a|b)\* | 以a为首字符的a、b字符串的集合 | {a , aa , ab , aab, aba , aaab , aaba , …} |
| (a|b)\*abb | 以abb结尾的a、b字符串的集合 | {abb, aabb, babb, abaabb, abaabb, …} |

【备考点拨】

1、掌握正规式相关的概念；

2、掌握常见的正规式表示含义。

### 2.4 有限自动机（★★）

【考法分析】

1、本知识点的主要考查形式有：给出一个确定或不确定的有限自动机，指出其能够识别的字符串，或指出对应的正规式表示。

【要点分析】

1、定义：M=(S,∑, δ,S0,Z)

1）S是一个有限集，每个元素为一个状态

2）∑是一个有穷字母表，每个元素为一个输入字符

3）δ是转换函数：是一个单值对照

4）S0,属于S，是其唯一的初态

5）Z是一个终态集（可空）

2、一个有限自动机所识别的语言是从开始状态到终止状态所有路径上的字符串的集合。要判断一个字符串能否被指定的自动机识别，就看在该自动机的状态图中能否找到一条从开始状态到达终止状态的路径，且路径上的字符串等于需要识别的字符串。而对于其正规式，可以通过能够识别的字符串去总结规律。

【备考点拨】

1、掌握有限自动机相关的基本概念；

2、掌握有限自动机能够识别的字符串判断；

3、掌握有限自动机与正规式的对应关系。

### 2.5 后缀表达式（★★★）

【考法分析】

1、本知识点的主要考查形式有：给出概念描述判断正误；给出表达式或其语法树，要求指出其后缀表达形式。

【要点分析】

1、**前缀表达式**（+ab）：把运算符写在运算对象的前面。（语法树的先序遍历）

2、**中缀表达式**（a+b）：把运算符写在运算对象的中间。

（语法树的中序遍历，是常见的表达式形式，父节点是运算符（括号不用表示），孩子结点是运算符的参数，据此可以构造出其语法树。）

3、**后缀表达式**（ab-）：又叫作**逆波兰式**，把运算符写在运算对象的后面。

（语法树的后序遍历，常与栈结合使用。）

【备考点拨】

1、掌握表达式构造语法树的过程；

2、掌握后缀表达式相关的概念和求取过程。

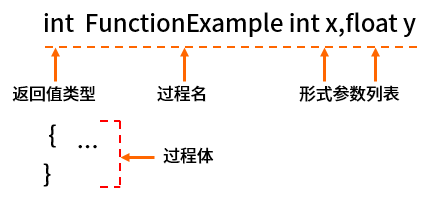
### 2.6 传值与传址（★★★★）

【考法分析】

1、本知识点的主要考查形式有：给出一些相关的描述判断其正误；给出一些代码，其中部分参数是传值调用，部分参数是传址调用，求取最终结果。

【要点分析】

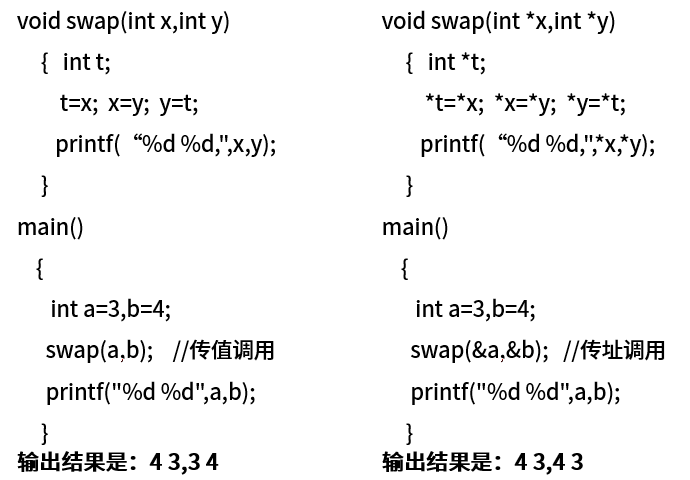
1、函数的格式：



2、传值和传址（引用）调用的对比：

|  |  |
| --- | --- |
| 传递方式 | 主要特点 |
| 传值调用 | 形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变 |
| 传址调用  引用调用 | 形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值 |

例：



【备考点拨】

1、掌握传值与传址调用相关概念；

2、能够根据代码，得出最终的结果（注意其中传值与传址调用的区别）。

### 2.7 多种程序语言特点（★★★）

【考法分析】

1、本知识点的主要考查形式有：给出特定描述指出其对应的程序语言；或给出程序语言指出其对应描述的正误。

【要点分析】

1、常见的程序设计语言：

Fortran语言（第一个高级程序设计语言，科学计算，执行效率高）

Pascal语言（结构化程序设计语言，表达能力强，Delphi）

C语言（通用、结构化程序设计语言，指针操作能力强，高效）

**Lisp语言**（函数式程序语言，符号处理，人工智能）

C++语言（C语言基础上增加了类机制，面向对象，高效，与C兼容）

Java语言（面向对象，中间代码，跨平台，通用的程序设计语言）

**Python**（面向对象，解释型程序设计语言，胶水语言，通用的脚本语言）

**PHP**（服务器端脚本语言，制作动态网页）

Ruby（简单快捷、面向对象、脚本语言）

Delphi（快速应用程序开发工具，可视化编程环境）

COBOL（数据处理领域最为广泛的程序设计语言，高级编程语言）

XML（可扩展标记语言，标准通用标记语言的子集 ）

PROLOG（逻辑式语言，间接性，表达能力强，建造专家系统、数据库、自然语言理解、智能知识库等）

注：C/C++常被用于操作系统开发；脚本语言是解释性语言。

【备考点拨】

1、掌握常用程序设计语言的特点，能够加以区分。

# 第10章 知识产权与标准化

## 1 考情分析

根据对历年的考试真题进行分析，本章要求考生掌握以下几个方面的知识：

（1）了解著作权法、软著权法、专利法、商标法、反不正当竞争法保护的范围与对象。

（2）了解相关知识产权的保护期限。

（3）区分职务作品以及合作开发，相关知识产权人的确定，了解商标与专利出现同时申请时产权人的确定方式。

（4）了解软著权的特殊性，了解软著权保护的对象和不保护的对象。

（5）了解不授予专利权、不适用著作权的一些特例，区分侵权行为与不侵权的行为。

（6）区分标准的分类，了解标准的编号特点。

### 1.1 本章重点

|  |  |  |
| --- | --- | --- |
| 序号 | 知识领域 | 知识点详情 |
| 1 | 保护对象和保护期限（★★） | 保护对象和保护期限 |
| 1 | 知识产权人确定（★★★） | 知识产权人确定 |
| 1 | 侵权判断（★★★★） | 侵权判断 |
| 1 | 标准的分类（★） | 标准的分类 |
| 1 | 标准代号的识别（★） | 标准代号的识别 |

## 2 知识点详情

### 2.1 保护对象和保护期限（★★）

【考法分析】

1、本知识点的主要考查形式是：给定描述判断其正误，判定哪些权利不需要申请、可以永久保护等。

【要点分析】

|  |  |  |
| --- | --- | --- |
| 法律法规名称 | 保护对象及范围 | 注意事项 |
| 著作权法 | 著作权  文学、绘画、摄影等作品 | 1、不需要申请，作品完成即开始保护  2、绘画或摄影作品原件出售（赠予）著作权还归原作者，原件拥有者有：所有权、展览权。 |
| 软件著作权法  计算机软件保护条例 | 软件著作权  软件作品 | 1、不需要申请，作品完成即开始保护  2、登记制度便于举证 |
| 专利法 | 专利权 | 需要申请，专利权有效期是从申请日开始计算 |
| 商标法 | 商标权 | 需要申请，核准之日起商标受保护 |
| 反不正当竞争法 | 商业秘密权 | 1、商业秘密包括技术与经营两个方面  2、必须有保密措施才能认定商业秘密 |

|  |  |  |
| --- | --- | --- |
| 客体类型 | 权力类型 | 保护期限 |
| 公民作品 | 署名权、修改权、保护作品完整权 | 没有限制 |
| 发表权、使用权和获得报酬权 | 作者终生及其死亡后的50年（第50年的12月31日） |
| 单位作品 | 发表权、使用权和获得报酬权 | 50年（首次发表后的第50年的12月31日），若其间未发表，不保护。 |
| 公民软件产品 | 署名权、修改权 | 没有限制 |
| 发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权 | 作者终生及死后50年（第50年12月31日）。合作开发，以最后死亡作者为准。 |
| 单位软件产品 | 发表权、复制权、发行权、出租权、信息网络传播权、翻译权、使用许可权、获得报酬权、转让权 | 50年（首次发表后的第50年的12月31日），若其间未发表，不保护 |
| 注册商标 | | 有效期10年（若注册人死亡或倒闭1年后，未转移则可注销，期满后6个月内必须续注） |
| 发明专利权 | | 保护期为20年（从申请日开始） |
| 实用新型和外观设计专利权 | | 保护期为10年（从申请日开始） |
| 商业秘密 | | 不确定，公开后公众可用 |

【备考点拨】

1、掌握著作权法、计算机保护条例（软件著作权法）、专利法、商标法、反不正当竞争法的保护对象和范围；

2、掌握著作权、软件著作权、专利权、商标权、商业秘密权的定义和保护期限。

### 2.2 知识产权人确定（★★★）

【考法分析】

1、本知识点的主要考查形式是：给定描述判断其正误。

【要点分析】

|  |  |  |  |
| --- | --- | --- | --- |
| 情况说明 | | 判断说明 | 归属 |
| 作品 | 职务  作品 | 利用单位的物质技术条件进行创作，并由单位承担责任的 | 除署名权外其他著作权归单位 |
| 有合同约定，其著作权属于单位 | 除署名权外其他著作权归单位 |
| 其他 | 作者拥有著作权，单位有权在业务范围内优先使用 |
| 软件 | 职务  作品 | 属于本职工作中明确规定的开发目标 | 单位享有著作权 |
| 属于从事本职工作活动的结果 | 单位享有著作权 |
| 使用了单位资金、专用设备、未公开的信息等物质、技术条件，并由单位或组织承担责任的软件 | 单位享有著作权 |
| 专利权 | 职务  作品 | 本职工作中作出的发明创造 | 单位享有专利 |
| 履行本单位交付的本职工作之外的任务所作出的发明创造 | 单位享有专利 |
| 离职、退休或调动工作后1年内，与原单位工作相关 | 单位享有专利 |

|  |  |  |  |
| --- | --- | --- | --- |
| 情况说明 | | 判断说明 | 归属 |
| 作品软件 | 委托创作 | 有合同约定，著作权归委托方 | 委托方 |
| 合同中未约定著作权归属 | 创作方 |
| 合作开发 | 只进行组织、提供咨询意见、物质条件或者进行其他辅助工作 | 不享有著作权 |
| 共同创作的 | 共同享有，按人头比例。  成果可分割的，可分开申请。 |
| 商标 | | 谁先申请谁拥有（除知名商标的非法抢注）  同时申请，则根据谁先使用（需提供证据）  无法提供证据，协商归属，无效时使用抽签（但不可不确定） | |
| 专利 | | 谁先申请谁拥有  同时申请则协商归属，但不能够同时驳回双方的专利申请 | |

【备考点拨】

1、掌握职务作品和个人作品的区分；

2、掌握合作开发和委托开发知识产权人的确定；

3、掌握相关知识产权人的确认方式。

### 2.3 侵权判断（★★★★）

【考法分析】

1、本知识点的主要考查形式是：给定描述判断其正误。

【要点分析】

1、中国公民、法人或者其他组织的作品，不论是否发表，都享有著作权。

2、开发软件所用的思想、处理过程、操作方法或者数学概念不受保护

3、著作权法不适用于下列情形：

（1）法律、法规，国家机关的决议、决定、命令和其他具有立法、行政、司法性质的文件，及其官方正式译文；

（2）时事新闻；

（3）历法、通用数表、通用表格和公式。

|  |  |
| --- | --- |
| 不侵权 | 侵权 |
| 个人学习、研究或者欣赏；  适当引用；  公开演讲内容  用于教学或科学研究  复制馆藏作品；  免费表演他人作品；  室外公共场所艺术品临摹、绘画、摄影、录像；  将汉语作品译成少数民族语言作品或盲文出版。 | 未经许可，发表他人作品；  未经合作作者许可，将与他人合作创作的作品当作自己单独创作的作品发表的；  未参加创作，在他人作品署名；  歪曲、篡改他人作品的；  剽窃他人作品的；  使用他人作品，未付报酬；  未经出版者许可，使用其出版的图书、期刊的版式设计的。 |

【备考点拨】

1、掌握常见的侵权行为和合理引用的行为。

### 2.4 标准的分类（★）

【考法分析】

1、本知识点的主要考查形式是根据题干描述或根据标准代号指出其分类。

【要点分析】

1、国际标准：ISO、IEC等国际标准化组织。

2、国家标准：GB—中国、ANSI—美国、BS—英国、JIS—日本。

3、区域标准：又称为地区标准，如PASC—太平洋地区标准会议、CEN—欧洲标准委员会、ASAC—亚洲标准咨询委员会、ARSO—非洲地区标准化组织。

4、行业标准：GJB—中国军用标准、MIT-S—美国军用标准、IEEE—美国电气电子工程师协会。（行业标准的辨认一般是排除其他标准分类）。

5、地方标准：国家的地方一级行政机构制订的标准。

6、企业标准

7、项目规范

【备考点拨】

1、掌握常见标准分类。

### 2.5 标准代号的识别（★）

【考法分析】

1、本知识点的主要考查形式是根据题干指出标准分类与标准代号的对应关系。

【要点分析】

1、国际、国外标准代号：标准代号+专业类号+顺序号+年代号。

2、我国国家标准代号：强制性标准代号为GB、推荐性标准代号为GB/T、指导性标准代号为GB/Z、实物标准代号GSB。

3、行业标准代号：由汉语拼音大写字母组成（如电子行业为SJ）（行业标准的辨认一般是排除其他标准分类）。

4、地方标准代号：由DB加上省级行政区代码的前两位。

5、企业标准代号：由Q加上企业代号组成。

【备考点拨】

1、掌握常见的标准代号，能够根据代号区分标准分类，或根据要求指出对应标准代号。