二级公共基础知识总结

**第一章 数据结构与算法**

1.1 算法  
算法：是指解题方案的准确而完整的描述。算法不等于程序，也不等计算机方法，程序的编制不可能优于算法的设计。

算法的特征包括：  
（1）可行性；  
（2）确定性，算法中每一步骤都必须有明确定义，不充许有模棱两可的解释，不允许有多义性；  
（3）有穷性，算法必须能在有限的时间内做完，包括合理的执行时间的含义；  
（4）拥有足够的情报。  
算法的基本要素：一是对数据对象的运算和操作；二是算法的控制结构。  
指令系统：一个计算机系统能执行的所有指令的集合。  
基本运算包括：算术运算、逻辑运算、关系运算、数据传输。  
算法的控制结构：顺序结构、选择结构、循环结构。  
算法基本设计方法：列举法、归纳法、递推、递归、减斗递推技术、回溯法。  
算法复杂度：算法时间复杂度和算法空间复杂度。 两个之间没有联系的。  
算法时间复杂度是指执行算法所需要的计算工作量。  
算法空间复杂度是指执行这个算法所需要的内存空间。  
**1.2 数据结构的基本基本概念**  
数据结构研究的三个方面：  
（1）数据集合中各数据元素之间所固有的逻辑关系，即数据的逻辑结构；  
（2）在对数据进行处理时，各数据元素在计算机中的存储关系，即数据的存储结构；  
（3）对各种数据结构进行的运算。  
数据结构是指相互有关联的数据元素的集合。  
数据的存储结构有顺序、链接、索引等。  
线性结构条件：  
（1）有且只有一个根结点；  
（2）每一个结点最多有一个前件，也最多有一个后件。  
非线性结构：不满足线性结构条件的数据结构。  
**1．3 线性表及其顺序存储结构**  
线性表是由一组数据元素构成，数据元素的位置只取决于自己的序号，元素之间的相对位置是线性的。  
在复杂线性表中，由若干项数据元素组成的数据元素称为记录，

而由多个记录构成的线性表又称为文件。

线性表的顺序存储结构具有以下两个基本特点：  
（1）线性表中所有元素的所占的存储空间是连续的；  
（2）线性表中各数据元素在存储空间中是按逻辑顺序依次存放的。  
**1．4 栈和队列**  
栈是限定在一端进行插入与删除的线性表。

1、先进后出 FILO;

1、支持子程序调用;

2、具有记忆功能；

3、可以不用顺序存放数据；

4、只能够在top首部进行操作，bottom是绝对不动的；

5、栈的存放数据的个数为 num = （bottom – top）+1；

队列是指允许在一端（队尾）进入插入，而在另一端（队头）进行删除的线性表。

1、Rear指针指向队尾，front指针指向队头。

3、先进先出FIFO,或者是后进后出LILO

2、循环队列里面的个数计算方法：

A、rear > front 的时候， num = rear – front；

B、rear < front 的时候， num = rear + n – front；  
**1．5 线性链表**  
数据结构中的每一个结点对应于一个存储单元，这种存储单元称为存储结点，简称结点。  
结点由两部分组成：（1）用于存储数据元素值，称为数据域；（2）用于存放指针，称为指针域，  
在链式存储结构中，存储数据结构的存储空间可以不连续，各数据结点的存储顺序与数据元素之间的逻辑关系可以不一致，而数据元素之间的逻辑关系是由指针域来确定的。  
链式存储方式即可用于表示线性结构，也可用于表示非线性结构。  
线性链表的基本运算：查找、插入、删除。  
**1．6 树与二叉树**  
树是一种简单的非线性结构，所有元素之间具有明显的层次特性。  
在树结构中，没有前件的结点只有一个，称为树的根结点，简称树的根。

每一个结点可以有多个后件，称为该结点的子结点。

没有后件的结点称为叶子结点。  
在树结构中，一个结点所拥有的后件的个数称为该结点的度，所有结点中最大的度称为树的度。树的最大层次称为树的深度。  
二叉树的特点：（1）非空二叉树只有一个根结点；（2）每一个结点最多有两棵子树，且分别称为该结点的左子树与右子树。  
二叉树的基本性质：必考的题目  
（1）在二叉树的第k层上，最多有2k-1(k≥1)个结点；  
（2）深度为m的二叉树最多有2m-1个结点；  
（3）度为0的结点（即叶子结点）总是比度为2的结点多一个；  
（4）二叉树中 n = n0 +n1 +n2

满二叉树是指除最后一层外，每一层上的所有结点有两个子结点，则k层上有2k-1个结点深度为m的满二叉树有2m-1个结点。  
完全二叉树是指除最后一层外，每一层上的结点数均达到最大值，在最后一层上只缺少右边的若干结点。  
二叉树存储结构采用链式存储结构，对于满二叉树与完全二叉树可以按层序进行顺序存储。  
二叉树的遍历：（一般画个图要你把顺序写出来）  
（1）前序遍历（DLR），首先访问根结点，然后遍历左子树，最后遍历右子树；  
（2）中序遍历（LDR），首先遍历左子树，然后访问根结点，最后遍历右子树；  
（3）后序遍历（LRD）首先遍历左子树，然后访问遍历右子树，最后访问根结点。  
**1．7 查找技术**  
顺序查找的使用情况：  
长度为n的线性表，找出一个数据，最差的情况为比较n 次。

长度为n的线性表，找出一个最大数据，最差的情况为比较n-1 次。  
二分法查找只适用于顺序存储的有序表，

二分查找：对于长度为n的有序线性表，最坏情况只需比较log2n次。  
**1．8 排序技术**  
排序是指将一个无序序列整理成按值非递减顺序排列的有序序列。  
交换类排序法：（1）冒泡排序法，需要比较的次数为n(n-1)/2；

（2）快速排序法，需要比较的次数为n(n-1)/2；。  
插入类排序法：（1）简单插入排序法，最坏情况需要n(n-1)/2次比较；

（2）希尔排序法，最坏情况需要O(n1.5)次比较。  
选择类排序法：（1）简单选择排序法, 最坏情况需要n(n-1)/2次比较；

（2）堆排序法，最坏情况需要O(nlog2n)次比较。

**第二章程序设计基础**

2．1 程序设计设计方法和风格  
注释分序言性注释和功能性注释，语句结构清晰第一、效率第二。不要弄反了。

程序一定要求具有 易读性，可读性较好。

程序设计方法有两种，结构化程序设计 和 面向对象程序设计。  
2．2 结构化程序设计  
结构化程序设计方法的四条原则是：考试重点都要背下来

1. 自顶向下；2. 逐步求精；3.模块化；4.限制使用goto语句。

注意，这四个特点中，最重要的是 模块化。

**结构化程序的基本结构**和特点：  
（1）顺序结构：一种简单的程序设计，最基本、最常用的结构；  
（2）选择结构：又称分支结构，包括简单选择和多分支选择结构，   
（3）循环结构：可根据给定条件，判断是否需要重复执行某一相同程序段。  
2．3 面向对象的程序设计  
面向对象的程序设计：以**对象**为核心。

面向对象具备哪些基本概念：

对象、类和实例、消息、继承、多态性、

面向对象方法的优点：  
（1）与人类习惯的思维方法一致；  
（2）稳定性好；  
（3）可重用性好；  
（4）易于开发大型软件产品；  
（5）可维护性好。  
对象是面向对象方法中最基本的概念，对象是实体的抽象。  
面向对象的程序设计方法中的对象是系统中用来描述客观事物的一个实体，是构成系统的一个基本单位，由一组表示其静态特征的属性和它可执行的一组操作组成。  
属性即对象所包含的信息，操作描述了对象执行的功能，操作也称为方法或服务。  
对象的基本特点：  
（1）标识惟一性；  
（2）分类性；  
（3）多态性； 重点  
（4）封装性； 重点  
（5）模块独立性好。  
类是指具有共同属性、共同方法的对象的集合。类是对象的抽象，对象是对应类的一个实例。  
消息是一个实例与另一个实例之间传递的信息。  
消息的组成包括：

1. 接收消息的对象的名称；（2）消息标识符，也称消息名；（3）零个或多个参数。

继承是指能够直接获得已有的性质和特征，而不必重复定义他们。  
继承分单继承和多重继承。对象也可以没有继承。

1、单继承指一个类只允许有一个父类，

2、多重继承指一个类允许有多个父类。  
多态性是指同样的消息被不同的对象接受时可导致完全不同的行动的现象

**第三章软件工程基础**

3．1 软件工程基本概念  
计算机软件是包括程序、数据及相关文档的完整集合。  
软件的特点包括：  
（1）软件是一种逻辑实体；  
（2）软件的生产与硬件不同，它没有明显的制作过程；  
（3）软件在运行、使用期间不存在磨损、老化问题；  
（4）软件的开发、运行对计算机系统具有依赖性，

（5）软件复杂性高，成本昂贵；  
（6）软件开发涉及诸多的社会因素。  
软件按功能分为**应用软件、系统软件、支撑软件（或工具软件）**。

1、数据库管理系统是系统软件。

2、教务处管理系统是应用软件。  
软件危机主要表现在**成本、质量、生产率**等问题。  
软件工程是应用于计算机软件的定义、开发和维护的一整套方法、工具、文档、实践标准和工序。  
软件工程包括3个要素：**方法、工具、过程**。  
方法：完成软件工程项目的技术手段。

工具：支持软件的开发、管理、文档生成。

过程：支持软件开发的各个环节的控制、管理。

软件工程过程是把软件转化为输出的一组彼此相关的资源和活动，包含4种基本活动：  
（1）P——软件规格说明；（2）D——软件开发；（3）C——软件确认；（4）A——软件演进。  
软件生命周期：软件产品从提出、实现、使用维护到停止使用退役的过程。  
软件生命周期三个阶段:软件定义、软件开发、运行维护，主要活动阶段是：  
需求分析是定义阶段。 详细设计和概要设计是开发阶段。  
软件工程的目标和与原则：  
目标：在给定成本、进度的前提下，开发出具有有效性、可靠性、可理解性、可维护性、可重用性、可适应性、可移植性、可追踪性和可互操作性且满足用户需求的产品。  
基本目标：付出较低的开发成本；达到要求的软件功能；取得较好的软件性能；开发软件易于移植；需要较低的费用；能按时完成开发，及时交付使用。  
基本原则：抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。  
软件工程的理论和技术性研究的内容主要包括：软件开发技术和软件工程管理。  
软件开发技术包括：软件开发方法学、开发过程、开发工具和软件工程环境。  
软件工程管理包括：软件管理学、软件工程经济学、软件心理学等内容。  
软件管理学包括人员组织、进度安排、质量保证、配置管理、项目计划等。  
软件工程原则包括抽象、信息隐蔽、模块化、局部化、确定性、一致性、完备性和可验证性。  
3．2 结构化分析方法  
结构化方法的核心和基础是结构化程序设计理论。  
需求分析方法有（1）结构化需求分析方法； （2）面向对象的分析的方法。  
需求分析阶段工作4个方面：

1、需求获取 2、需求分析 3、编写需求规格说明书（重点） 4、需求评审

从需求分析建立的模型的特性来分：静态分析和动态分析。  
结构化分析方法的实质：面向数据流，自顶向下，逐层分解，建立系统的处理流程，以数据流图和数据字典为主要工具,建立系统的逻辑模型。  
结构化分析的常用工具：  
（1）数据流图； （2）数据字典； （3）判定树； （4）判定表。  
数据流图：描述数据处理过程的工具，是需求理解的逻辑模型的图形表示，它直接支持系统功能

建模。  
数据字典：对所有与系统相关的数据元素的一个有组织的列表，以及精确的、严格的定义，使得

用户和系统分析员对于输入、输出、存储成分和中间计算结果有共同的理解。

数据字典是结构化分析的核心。  
软件需求规格说明书的特点：**（核心的是无歧义性）**（1）正确性；（2）无岐义性；（3）完整性；（4）可验证性；

（5）一致性；（6）可理解性；（7）可追踪性。  
3．3 结构化设计方法  
软件设计的基本目标是用比较抽象概括的方式确定目标系统如何完成预定的任务，

软件设计是确定系统的物理模型。  
软件设计是开发阶段最重要的步骤，是将需求准确地转化为完整的软件产品或系统的唯一途径。  
从技术观点来看，软件设计包括软件结构设计、数据设计、接口设计、过程设计。。  
从工程管理角度来看：概要设计和详细设计。  
软件设计的一般过程：软件设计是一个迭代的过程；先进行高层次的结构设计；后进行低层次的过程设计；穿插进行数据设计和接口设计。  
衡量软件模块独立性使用耦合性和内聚性两个定性的度量标准。

耦合性：对一个软件结构中不同模块之间互连程度的度量。

内聚性：一个模块内部各个元素间彼此结合的紧密程度的度量。  
在程序结构中各模块的内聚性越强，则耦合性越弱。优秀软件应高内聚，低耦合。  
软件概要设计的基本任务是：  
（1）设计软件系统结构； （2）数据结构及数据库设计；  
（3）编写概要设计文档； （4）概要设计文档评审。  
模块用一个矩形表示，箭头表示模块间的调用关系。   
在结构图中还可以用带注释的箭头表示模块调用过程中来回传递的信息。还可用带实心圆的箭头表示传递的是控制信息，空心圆箭心表示传递的是数据。  
结构图的基本形式：基本形式、顺序形式、重复形式、选择形式。  
结构图有四种模块类型：传入模块、传出模块、变换模块和协调模块。  
典型的数据流类型有两种：变换型和事务型。  
变换型系统结构图由输入、中心变换、输出三部分组成。  
事务型数据流的特点是：接受一项事务，根据事务处理的特点和性质，选择分派一个适当的处理单元，然后给出结果。  
详细设计：是为软件结构图中的每一个模块确定实现算法和局部数据结构，用某种选定的表达工具表示算法和数据结构的细节。  
常见的过程设计工具有：**考试重点**

1、图形工具（程序流程图）、 2、表格工具（判定表）、 3、语言工具（PDL）。

程序流程图是考试重点，**其中->箭头表示控制流，<>菱形表示逻辑条件**。  
3．4 软件测试  
软件测试定义：使用人工或自动手段来运行或测定某个系统的过程，其目的在于检验它是否满足规定的需求或是弄清预期结果与实际结果之间的差别。  
软件测试的目的：发现错误而执行程序的过程。  
软件测试方法：静态测试和动态测试。  
静态测试：包括代码检查、静态结构分析、代码质量度量。不实际运行软件，主要通过人工进行。  
动态测试：是基本计算机的测试，主要包括白盒测试方法和黑盒测试方法。  
白盒测试：在程序内部进行，主要用于完成软件内部CAO作的验证。主要方法有逻辑覆盖、基

本基路径测试。  
黑盒测试：主要诊断功能不对或遗漏、界面错误、数据结构或外部数据库访问错误、性能错误、

初始化和终止条件错，用于软件确认。

白盒主要方法：1、逻辑覆盖测试 2、基本路径测试。

黑盒主要方法：1等价类划分法、2边界值分析法、3错误推测法、4因果图。  
软件测试过程一般按4个步骤进行（要背先后顺序）：

单元测试、集成测试、验收测试（确认测试）和系统测试。  
3．5 程序的调试  
程序调试的任务是诊断和改正程序中的错误，主要在开发阶段进行。  
软件调试可分表静态调试和动态调试。

静态调试主要是指通过人的思维来分析源程序代码和排错，是主要的设计手段，

动态调试是辅助静态调试。

调试方法有：（1）强行排错法；（2）回溯法；（3）原因排除法。

**第四章 数据库设计基础**

4．1 数据库系统的基本概念  
数据：实际上就是描述事物的符号记录。  
数据的特点：有一定的结构，有型与值之分，如整型、实型、字符型等。

数据库：长期存储在计算机内部的、有组织、可共享的数据的集合。数据库存放数据是按数据所提供的数据模式存放的具有**集成**与**共享**的特点。

数据库管理系统：一种系统软件，负责数据库中的数据组织、数据操纵、数据维护、控制及保

护和数据服务等，是数据库的核心。

数据库管理系统提供以下的数据语言：  
（1）数据定义语言：负责数据的模式定义与数据的物理存取构建；  
（2）数据操纵语言：负责数据的操纵，如查询与增、删、改等；  
（3）数据控制语言：负责数据完整性、安全性的定义与检查以及并发控制、故障恢复等。  
数据语言按其使用方式具有两种结构形式：交互式命令(又称自含型或自主型语言)宿主型语言（一般可嵌入某些宿主语言中）。  
数据库管理员：对数据库进行规划、设计、维护、监视等的专业管理人员。  
数据库系统：由数据库（数据）、数据库管理系统（软件）、数据库管理员（人员）、硬件平台（硬

件）、软件平台（软件）五个部分构成的运行实体。DBS是最大的包括其他所有的。  
数据库应用系统：由数据库系统、应用软件及应用界面三者组成。  
数据库发展的三个阶段：人工管理阶段、文件系统阶段、数据库管理阶段。  
数据库系统阶段的基本特点：

1、数据的集成性 、

2、数据的**高共享性与低冗余性** 、

3、数据独立性（物理独立性与逻辑独立性）、数据统一管理与控制。  
数据库系统的三级模式：  
（1）概念模式：数据库系统中全局数据逻辑结构的描述，全体用户公共数据视图；  
（2）外模式：也称子模式与用户模式。是用户的数据视图，也就是用户所见到的数据模式；  
（3）内模式：又称物理模式，它给出了数据库物理存储结构与物理存取方法。

一个数据库只有一个内模式，可以有多个外模式。  
数据库系统的两级映射：  
（1）概念模式到内模式的映射；  
（2）外模式到概念模式的映射。  
4.2 数据模型  
数据模型的概念：是数据特征的抽象，从抽象层次上描述了系统的静态特征、动态行为和约束条件，为数据库系统的信息表与操作提供一个抽象的框架。描述了数据结构、数据操作及数据约束。  
E-R模型的基本概念  
（1）实体：现实世界中的事物；  
（2）属性：事物的特性；  
（3）联系：现实世界中事物间的关系。实体集的关系有一对一、一对多、多对多的联系。  
E-R模型的图示法：（1）实体集表示法； （2）属性表法； （3）联系表示法。

E-R模型的:矩形表示实体；椭圆表示矩形；棱形表示联系。  
关系模型采用二维表来表示，简称表，由表框架及表的元组组成。

1、一个二维表就是一个关系。  
2、在二维表中凡能唯一标识元组的最小属性称为键或码。具有唯一性。

3、元组就是二维表中一行，属性就是二维表表中一列。

4、从所有侯选健中选取一个作为用户使用的键称主键。

5、表A中的某属性是某表B的键，则称该属性集为A的外键或外码。

6、分量是不可以再分的。

以上六点非常重要。  
关系中的数据约束：  
（1）实体完整性约束：约束关系的主键中属性值不能为空值；  
（2）参照完全性约束：是关系之间的基本约束；  
（3）用户定义的完整性约束：它反映了具体应用中数据的语义要求。

4.3关系代数  
关系数据库系统的特点之一是它建立在数据理论的基础之上，有很多数据理论可以表示关系模型的数据操作，其中最为著名的是关系代数与关系演算。  
关系模型的基本运算：  
（1）插入 （2）删除 (3)修改 （4）查询（包括**投影、选择、笛卡尔积**）  
两个表的运算一般会考到：投影、链接、选择、笛卡尔积、交、并当中的一个。

4.4 数据库设计与管理  
数据库设计是数据应用的核心。  
数据库设计的两种方法：  
（1）面向数据：以信息需求为主，兼顾处理需求；  
（2）面向过程：以处理需求为主，兼顾信息需求。  
数据库的生命周期：需求分析阶段、概念设计阶段、逻辑设计阶段、物理设计阶段、编码阶段、测试阶段、运行阶段、进一步修改阶段。  
需求分析常用结构析方法和面向对象的方法。结构化分析（简称SA）方法用自顶向下、逐层分解的方式分析系统。用数据流图表达数据和处理过程的关系。对数据库设计来讲，数据字典是进行详细的数据收集和数据分析所获得的主要结果。  
数据字典是各类数据描述的集合，包括5个部分：数据项、数据结构、数据流（可以是数据项，也可以是数据结构）、数据存储、处理过程。  
数据库概念设计的目的是分析数据内在语义关系。

**公共基础补充知识点**

1. 一个数据结构的逻辑结构根据需要可以有多个存储结构。存储结构的不同，会造成处理

的效率不同。

2、程序设计方法和技术的发展经过了结构化程序设计和面向对象设计两个阶段。

3、当今程序设计的风格是“清晰第一，效率第二”。

4、程序可以没有输入，但是一定要有输出。

**5、结构化程序设计遵循：自顶向下，逐步求精，模块化，限制使用goto语句（常考）**。

6、面向对象的基本特点：标志唯一性，分类性，多态性，封装性，模块独立性。尤其重要的是多态性和封装性。没有类比性。

**7、多态性**：统一操作可以是不同对象的行为。同样的消息被不同的对象接收时可导致不同的动作的现象。

8**、封装性**：从外面看不到对象的内部，只能看到对象的外部特性。

9、类：是具有共同属性、共同方法的对象的集合。描述了属于该对象类型的所有对象的性质，而一个对象则是对应类的一个实例。（**常考**）

10、消息：是指对象间的相互合作的协作机制，是一个对象与另一个对象之间的传递的消息。

11继承：是指使用已有的类定义作为基础建立新类的定义技术。继承分为单继承和多继承。

12、面向对象中，类的实例叫做对象。

13、源程序文档化要求程序应该加上注释。注释一般为序言性注释和功能性注释。

14、面向对象方法和技术是以对象为核心。

**软件工程基础**

1. 软件工程的核心思想是把软件当作一个工程产品来处理。
2. 软件开发的三个阶段以及每个阶段的任务：

可行性研究初步项目计划

需求

分析

概要

设计

详细

分析

实现

测试

使用

维护

退役

定义阶段　　　　　　　　　　　开发阶段　　　　　　　　　　　维护阶段

　　　　　　这个表请大家抽时间背下。软件开发的三个阶段，每个阶段的工程。

1. 软件开发方法包括分析方法，设计方法，程序设计方法。
2. 结构化方法包括结构化分析方法，结构化设计方法，结构化编程方法。
3. 结构化分析方法在软件需求分析阶段的应用。
4. 结构化分析常用的工具中最重要的工具是数据流图。　　　表示加工，　　表示数据流，

表示数据存储（数据源） 表示源和潭 。

1. 软件规格说明书（SRS）是需求分析阶段的最后结果，是软件开发文档重要的文档之一。
2. 软件设计要求高耦合性和低内聚性是考试的重点,要知道什么是耦合,什么是内聚。
3. 软件测试和程序调试的区别。软件测试的过程涵盖了整个软件生命周期。目的是为了发现错误而执行程序的过程。

10、从是否要执行被测试软件的角度，软件测试分为静态测试和动态测试。

11、从功能上划分，软件测试分为白盒测试和黑盒测试。

12、白盒测试是检查内部成分，方法有：逻辑覆盖测试方法，基本路径测试。

13、黑盒测试是对软件已经实现的功能是否满足需求进行测试和验证。方法有：等价类划分法，边界值划分法，错误推测法。

1. 程序调试的任务：诊断和改正程序中错误，它和软件测试不同，调试主要在开发阶段。
2. 软件调试方法：强制排错法，回朔法，原因排除法。
3. 软件维护不属于软件生命周期开发阶段的任务。
4. 软件进行了程序调试后还要进行测试。
5. 软件工程的主要思想是：强调在软件开发过程中需要应用工程化的原则。
6. 软件设计中，不属于过程设计工具的是：DFD图。
7. 结构化分析常见的工具：DFD图，DD（数据字典），判定树，判定表。
8. 程序和软件是不同的，软件是程序、数据与相关文档的集合。
9. 软件的开发、运行对计算机系统具有依赖性。
10. 软件工程的三要素：工具、过程、方法。
11. 软件生命周期中，能够准确确定软件系统必须做什么和必须具备哪些功能的阶段是：需求分析。
12. 数据流图只含有数据流，不含有控制流。数据流图用于抽象描述一个软件的逻辑模型。
13. 数据流的类型有两种：事务型和变换型。在很多软件应用中，存在某种作业数据流，它可以引发一个或多个处理，这些处理能够完成该作业要求的功能，这种数据流叫做事务型数据流。
14. 程序流程图中的箭头代表的是：控制流。
15. 在结构化方法中，软件功能分解属于总体设计阶段。
16. 内聚型中内聚性最强的是：功能内聚。
17. 检查软件产品是否符合需求定义的过程是：确认测试。
18. 软件危机：泛指在计算机软件的开发和维护过程中遇到的一系列严重的问题。
19. Jackson 方法是一种面向数据结构的结构化方法。
20. 软件开发环境是全面支持软件开发全过程的软件开发工具集合。
21. 软件需求是指用户对目标软件系统在功能、行为、性能设计约束等方面的期望。
22. 为每一个模块确定实现算法和局部数据结构的是详细设计阶段的任务。

**数据库设计基础**

1. 关系运算是考试的重点，考的多的是并和笛卡尔积。不改变属性但减少元组个数的是并。投影也是比较喜欢考的，一般的意思是从4、5个列中选出2、3个。笛卡尔积则是会增加元组的个数。
2. E-R图中，矩形、菱形、椭圆形等代表的意思是考试的重点。
3. 数据库系统中，数据的物理结构不一定要和逻辑结构一致。
4. 数据库技术的根本目标是解决数据的共享问题。
5. 数据库系统中，用户所见的数据模式为：外模式。
6. 数据库设计的四个阶段是：需求分析、概念设计、逻辑设计和物理设计。
7. 两个实体间的关系是：一对一，多对多，一对多等，这种题目根据具体的题目分析。（常考题目）
8. 数据库DB，数据库系统DBS，数据库管理系统DBMS之间的关系：DBS包含DB和DBMS。
9. 数据库系统的核心：数据库管理系统。

10、数据独立性：是指数据与程序的互不依赖性。他包括数据的物理独立性和逻辑独立性。物理独立性是数据的物理结构（包括存储结构、存取方式）的改变，不影响数据库的逻辑结构。

11、数据的逻辑独立性是指数据库总体逻辑结构的改变，不需要相应修改应用程序。

12、用树形结构表示实体之间联系的模型是：层次模型。

13、数据库系统中修改数据模式、增加新的数据类型、改变数据间的联系等，不需要相应修改应用程序这一特点是逻辑独立性。

14、不涉及具体的硬件环境和平台，也与具体的软件环境无关的模式是：概念模型。

15、用户数据是建立在外模式的框架上的。

16、给出数据的全局逻辑结构到数据的物理存储结构间的对应关系的映射是：概念模式到内模式的映射。

17、网状模型：不加任何的限制的无向图。

18、不准关系应用不存在的元组的约束是：参照完整性约束。

19、数据库设计包括两个方面的设计内容，它们是：概念设计和逻辑设计。

20、数据库设计中的主流方法是：面向数据的设计方法。

21、数据库概念设计过程分三个步骤进行：首先选择局部应用，在局部视图设计，最后进行视图集成。

22、数据库系统中，实现各种数据管理功能的核心软件称为数据库管理系统。

23、一个关系表的行称为元组。

24、在关系模型中，数据看成是二维表，每一个二维表称为一个关系。

25、数据管理技术经历了三个阶段：人工管理、文件系统和数据库系统。

26、数据库的基本特点：数据的集成性、**数据的高共享性和低冗余性（常考）**、**数据的独立性（常考）**、数据统一管理和控制。

27、三级模式中反映用户对数据的要求的模式是：数据模型。

28、物理数据模型是数据特征的抽象，他从抽象的层次上描述了系统的静态特征、动态行为和约束条件。

29、E-R图由实体、联系、属性三个部分组成。

30、二维表中能够唯一标识元组的最小属性集称为该表的码。

31、要求关系的主键中属性不能为空的约束是：实体完整性约束。

32、数据库的设计方法有两种，一种是面向数据的设计方法，一种是面向过程的设计方法。