ISO/IEC软件质量模型

6大质量特性 21个子特性

1. 功能性（软件所具有的的各项功能及规定性质的属性）

适合性：能否提供一组与功能适合程度有关的软件。

准确性：能否得到正确的软件属性（属性包含准确程度）

互用性：与其他制定系统的交互能力的软件属性。

依从性：使软件遵循有关规定和标准的属性。

安全性：防止非授权的故意以及意外访问的能力属性。

1. 可靠性（在规定运行条件下和规定运行周期下，软件维护性能级别的能力）

成熟性：由软件故障引起失效的频度有关的软件属性。

容错性：出现软件故障和违反指定接口，依然维持规定性能水平的属性。

可恢复性：失效发生后，重建性能水平，并恢复受影响的数据的能力属性。

1. 可用性（用户对使用软件所需要的能力程度的评价）

可理解性：用户理解软件概念及应用范围所做的努力。

易学性：用户应用软件所做的努力。

可操作性：用户操作和运行控制所做的努力。

（4）效率（规定条件下，软件性能级别与所用资源总量的关系能力属性。）

时间特性：软件执行时，响应及处理时间和吞吐量的属性。

资源特性：软件执行时，软件所使用的资源数量和使用资源时间的属性。

（5）可维护性（对软件修改的难易程度的能力属性）

可分析性：诊断失效原因所需要做的努力。

可改变性：修改 排错 适应环境所做的努力。

可测试性：与确认已修改软件所做的努力。

稳定性：修改所造成的未预料结果风险属性。

（6）可移植性（将软件从一个环境移到另一个环境）

适应性：适应不同环境。

可安装性：在指定环境下安装所做的努力。

遵循性（一致性）：遵循有关规定和约定。

可替换性：替代其它指定软件的机会。

改正在软件系统开发阶段已经发生而系统测试阶段还没有发现的错误，属于正确性维护。

某企业由于外部市场环境和管理需求的变化对现有软件系统提出新的需求，则对该软件系统进行的维护属于适应性维护。

某银行为了使其网上银行系统能够支持信用卡多币种付款功能而进行扩充升级，这需要对数据类型稍微进行一些改变，这一状况需要对网上银行系统进行适应性维护。

在软件维护阶段，为软件的运行增加监控设施属于完善性维护。

一软件在故障发生后，要求在90秒内恢复其性能和受影响的数据，与达到此目的有关的软件属性为易恢复性子特性。

软件可靠性是指一个系统在给定时间间隔内和给定条件下无失效运行的概率。

设计模式分为 创建型 结构型 行为性。采用设计模式是为了复用成功的设计。

创建型：工厂方法 抽象工厂 单例 生成器 构建 原型

结构型：适配器 合成 装饰 代理 享元 门面 桥接

行为型: 模板 迭代器 责任链 命令 备忘录 状态 访问者 解释器 调停者 观察者。

观察者设计模式（Observer）定义对象间一对多的依赖关系，当一对象发生改变时，所有依赖它的对象都将发生改变。并自动更新。

Adapter适配器模式将一个类的接口转换成客户希望的另外一个接口。A d a p t e r模式使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

门面模式（外观模式）提供一个接口访问子系统多个不同的接口。

装饰模式的意图是动态的给一个对象增加职责，不改变原类文件和使用继承的情况下，动态扩展一个对象的功能。在不影响其他对象的前提下，以动态透明的方式给单个对象添加职责。

迭代器设计模式（Iterator）的意图是提供一种顺序访问一个聚合对象中的各个元素，而不暴露该对象内部表示。

单例模式确保一个类只有一个实例，自行实例化，并向整个系统提供这个实例。

享元模式的意图是运用共享技术有效的支持大量细粒度的对象。

策略模式的意图是定义一系列的算法，把它们封装起来，并且使他们可以相互替换。本模式使得算法独立于使用它的客户而变化。

Abstract Factory（抽象工厂模式）：提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

Command（命令模式）：将一个请求封装为一个对象，从而使你可用不同的请求对客户进行[参数化](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=38199498&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)；对请求排队或记录请求日志，以及支持可取消的操作。

Chain of [Responsibility](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=1654456&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)（职责链模式）：为解除请求的发送者和接收者之间耦合，而使多个对象都有机会处理这个请求。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它。

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。这样以后就可将该对象恢复到保存的状态。

设计模式Factory Method(工厂模式)：定义一个用于创建对象的接口，让子类决定将哪一个类实例化。Factory Method使一个类的实例化延迟到其子类。

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。它使你可以在不改变各元素的类的前提[下定义](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7705050&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)作用于这些元素的新操作。

桥接模式意图是将抽象部分与实现部分分离，独立的变化。适用于

（1）避免抽象方法和实现方法捆绑一起。

（2）类的抽象以及它的实现都应该可以通过生成子类的方法扩充。

（3）对一个抽象的实现方法的修改应对客户不影响，对应客户的代码不变。

（4）想在多个对象间共享实现，要求客户不知道

代理模式的意图是为其它对象提供一个代理或地方以控制对这个对象的访问。

Composite(组合模式)：将对象组合成[树形结构](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7714336&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)以表示“部分-整体”的层次结构。它使得客户对单个对象和复合对象的使用具有一致性。描述了如何构造一个类层次式结构。

Proxy（代理模式）：为其他对象提供一个代理以控制对这个对象的访问。通过提供与对象相同的接口来控制对这个对象的访问

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。对象看起来似乎修改了它所属的类。

---------------------------------------------------------------------------------------------------------------------------------

Builder（建造者模式）：将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。

Interpreter（解析器模式）：给定一个语言, 定义它的文法的一种表示，并定义一个解释器, 该解释器使用该表示来解释语言中的句子。

Mediator（中介模式）：用一个中介对象来封装一系列的对象交互。中介者使各对象不需要显式地相互引用，从而使其耦合松散，而且可以独立地改变它们之间的交互。

[Prototype](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=45373&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)（原型模式）：用原型实例指定创建对象的种类，并且通过拷贝这个原型来创建新的对象。

Template Method（模板方法模式）：定义一个操作中的算法的骨架，而将一些步骤延迟到子类中。Template Method使得子类可以不改变一个算法的结构即可[重定义](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=73746426&ss_c=ssc.citiao.link" \t "https://baike.sogou.com/_blank)该算法的某些特定步骤。

对称加密技术(包括DES 3DES RC-5 IDEA)

DES 该算法采用替换和移位方式 对64位数据加密，秘钥长度56位。速度快，秘钥易产生。

3DES 三重DES 两个秘钥。加密：K1加密 K2解密 K1加密

解密：K1解密 K2加密 K1解密 秘钥长度加倍112位

RC-5 RSA数据安全公司多数产品使用。

IDEA 64位数据块，128位秘钥。比DES加密性好。

非对称加密技术（RSA RSC ECC）

RSA 属于公钥密码系统，1024位秘钥 反向使用公钥密码系统便提供了数字签名的功能，RSA 的安全性较高，但是随之而来的计算量也较大（例如指数运算），且会随着规模的增加而增大。RSA和DSA的安全性及其它各方面性能都差不多，而ECC较之则有着很多的性能优越，包括处理速度，带宽要求，存储空间等等。ECC 160位秘钥

对称加密称为私钥加密，对称密码编码技术。加密高效，但加密强度不大，秘钥分发困难。

非对称加密成为公钥加密，一个公钥对应一个私钥，加强加密强度，安全性提升，但加密速度慢，为对称加密的数千倍。经常加密信息摘要，（数据量小的）。

---------------------------------------------------------------------------------------------------------------------------------

信息摘要（MD5 SHA）

信息摘要简要描述较长的信息或文件。可以用于数字签名。

SHA 散列值160位（加密密钥长度） 安全性较高。

MD5 散列值128位（加密密钥长度）

UML常见图

结构性视图（静态）结构领域主要是对系统中的结构成员及相互关系进行描述

行为性视图（动态）行为领域则描述系统随时间变化的行为。

---------------------------------------------------------------------------------------------------------------------------------

结构性视图

（核心）类图：类图描述一组类 接口 协作和他们之间的关系。类图给出系统的静态设计视图，活动类图给出系统的静态进程视图。

对象图：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。这些图给出系统的静态设计图或静态进程视图。但对象图是从真实案例或原型案例角度建立。

包图：包图描述由模型本身分解而成的组织单元，以及他们之间的依赖关系。

组合结构图：组合结构图描述结构化类（构件或类）的内部结构，包括结构化类与系统其余部分的交互点。组合结构图用于画出结构化类的内部内容。

构件图：构件图描述一个封装的类和它的接口 端口 以及内嵌的构件和连接构件的内部结构。构件图用于表示系统的静态设计实现视图。构件图是类图的变体。

部署图：部署图描述对运行时的处理结点及在其中生存的构件配置，部署图给出了架构的静态部署视图，通常一个结点包括一个或多个部署图。

制品图：制品图描述计算机中一个系统的物理结构，制品包括文件 数据库 和类似的物理比特集合，制品图经常和部署图一起使用，制品也给出它们实现的类和构件。

行为性视图

（核心）用例图：通常描述一组用例 参与者与它们之间的关系。用例图给出系统的静态用例视。

这些图对系统的行为进行组织和建模时非常重要。

（核心）顺序图：顺序图一种交互图，交互图展现一种交互，由一组对象或参与者以及他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。顺序图强调消息的时间次序的交互图。

通信图：通信图是一种交互图，强调收发消息的对象或参与者的结构组织。该图反映了对象之间的消息交互，与顺序图相似但不同，协作图不但描述了对象之间的交互还描述交互对象之间的链接关系。，即通信图同时反映了系统的动态和静态特征。通信图（是协作图）

定时图：定时图也是一种交互图，强调消息跨越不同对象或参与者的实际时间，而不是仅仅只关心消息的相对时间。

状态图：状态图描述一个状态机，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，它对接口 类或协作的行为建模很重要。它强调事件导致的对象行为，有助于反应式系统建模。

活动图：活动图将进程或其他计算机结构展示为计算机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，对系统的功能建模和业务流程建模很重要，强调对象间控制流程。

交互概览图：交互概览图是活动图和顺序图的混合物。

加X锁后  不可以加任何锁  可读可改  
加S锁后  只能加S锁         可读不可改

工作项目毫无章法，没有管理。凭借项目负责人经验。（1）已执行级

建立基本的项目管理和实践来跟踪项目费用、进度和功能特性（2）已管理级

组织具有标准软件过程（3）已定义级（标准级）

对软件过程和产品都有定量的理解和控制（4）量化管理级

先进的新思想和新技术促进过程不断改进（5）优化级

时间复杂度

1. 顺序排序平均O（n） 冒泡和选择排序递归排序平均O（n2） 其它平均为O（nlogn）
2. 选择 堆 归并 基数排序最好最坏与平局一样。
3. 插入 希尔 冒泡 最好O（n） 平均O（nlogn） 最坏O（n2）
4. 快速排序 最好O（nlogn）平均O（nlogn） 最坏O（n2）
5. 希尔 选择 堆 快速 不稳定。
6. 选择排序同值从前到后优先。
7. 冒泡排序同值从后到前优先。

分治法是一般自顶向下的子问题求解

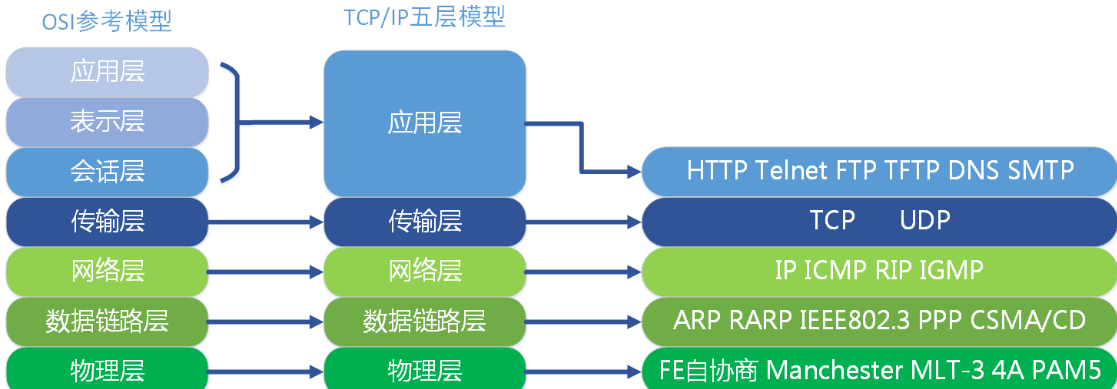
快速排序 归并排序属于分治法。

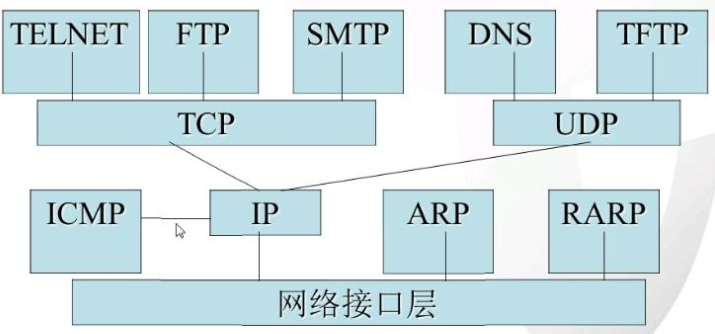
动态规划算法是一般自底向上的子问题重叠求解，以最优解为目标。

贪心算法是当前最优解（集装箱算法）

回溯法是深度优先算法，分支限界法是广度优先算法。

|  |  |  |  |
| --- | --- | --- | --- |
|  | 算法 | 时间复杂度 | 空间复杂度 |
| 矩阵问题 | 动态规划法 | O（n3） | O（n2） |
| 归并问题 | 分治法 |  |  |





33 内聚（内聚依次降低）与耦合（耦合依次升高）

功能内聚：完成单一功能，各部分协同合作，缺一不可。

顺序内聚：必须按顺序执行（比如依赖）。

通信内聚：所有处理元素集中于一个数据结构区域。

过程内聚：必须按照特定的次序进行。

瞬时内聚：所有任务同一时间间隔内完成。

逻辑内聚：逻辑上相关的一组任务（几个程序块内有相同部分，块内语句无联系）

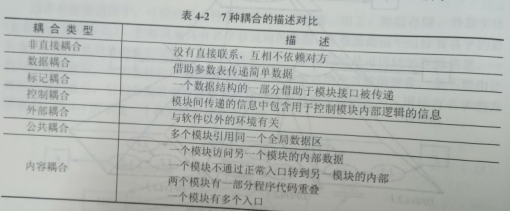
模块A执行几个逻辑上相似的功能，通过参数确定该模块完成哪一个功能，则该模块具有逻辑内聚。偶然内聚：（巧合内聚）偶然完成一组没有关系或松散关系的任务。

原则上一个模块的作用范围应该在其控制范围之内，若没有，

则可以将判定所在模块合并到父模块中，使判定处于较高层次。

也可以将受判定影响的模块下移到控制范围

也可以将判定上移到较高层次。



直接耦合：两个模块之间没有直接关系，它们之间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：一个模块访问另一个模块时，彼此之间是通过简单数据参数 (不是控制参数、公共数据结构或外部变量) 来交换输入、输出信息的。  
标记耦合 ：一组模块通过参数表传递记录信息，就是标记耦合。这个记录是某一数据结构的子结构，而不是简单变量。其实传递的是这个数据结构的地址；  
控制耦合：如果一个模块通过传送开关、标志、名字等控制信息，明显地控制选择另一模块的功能，就是控制耦合。  
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，则称之为外部耦合。  
公共耦合：若一组模块都访问同一个公共数据环境，则它们之间的耦合就称为公共耦合。

内容耦合：如果发生下列情形，两个模块之间就发生了内容耦合

(1) 一个模块直接访问另一个模块的内部数据；  
(2) 一个模块不通过正常入口转到另一模块内部；  
(3) 两个模块有一部分程序代码重叠(只可能出现在汇编语言中)；  
(4) 一个模块有多个入口。  
本题属于数据耦合，采购子系统模块给财务子系统模块传递数据。

软件设计原则：高内聚，低耦合。内聚是指模块内部各元素之间的联系紧密程度，表现为功能的好坏。耦合是指模块之间相互联系的紧密程度。

SQL

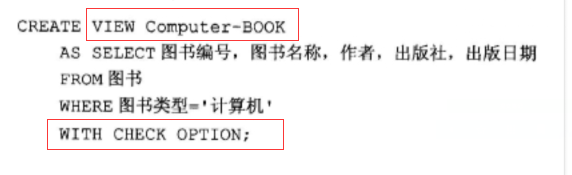
1 要将部门表Dept中name列的修改权限赋予用户Ming，并允许Ming将该权限授予他人。实现该要求的SQL语句如下：  
GRANT UPDATE(name) ON TABLE Dept TO Ming (51)；

答案：WITH GRANT OPTION

2 将Students表的插入权限赋予用户UserA，并允许其将该权限授予他人，应使用的SQL语句为： GRANT (51) TABLE Students TO UserA (52);

答案：INSERT ON；WITH GRANT OPTION

3建立“计算机”类图书的视图Computer-BOOK, 并要求进行修改、插入操作时保证该视图只有计算机类的图书。实现上述要求的SQL语句如下：



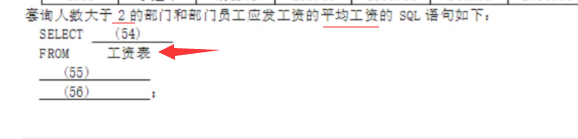
WITH CHECK OPTION 表示对UPDATE，INSERT，DELETE操作时保证更新、插入或删除的行满足视图定义中的谓词条件(即子査询中的条件表达式)。另外，组成视图的属性列名或者全部省略或者全部指定。

视图：在数据库系统中，视图是一个虚拟表，查询时从一个或多个基本表导出。

答案： 

对比WITH GRANT OPTION WITH CHECK OPTION

4 GROUP使用



答案：；

WHERE表示对表的限定，如下题。在GROUP BY子句后面跟一个HAVING子句可以对元组在分组前按照某种方式加上限制。但HAVING COUNT(DISTINCT(部门)加DISTINCT结果一定等于1（后边任何结果都是不可以的）

5SELECT (52) FROM (53) WHERE (54);

答案：A,R.C,F,G ；R，S；R.B=S.B AND R.C=S.C

对比WHERE后边的GROUP 与HAVING COUNT

软件测试的分类

动态测试是通过运行程序发现错误。对软件产品进行动态测试时可以采用黑盒测试和白盒测试法。黑盒测试法在完全不考虑软件的内部结构和特性的情况下，测试软件的外部特性

（1）白盒测试：（结构测试）用于单元测试阶段，把程序封装于透明的盒子里，测试者完全知道程序的结构及处理算法。应根据程序的内部逻辑和指定的覆盖标准确定测试数据。

（2）黑盒测试：（功能测试）用于集成测试和确认测试阶段，封装于不透明盒子，完全不了解部程序的结构及处理算法。只检查软件功能是否按照需求说明书的要求正常使用

（3）α测试：是用户在开发者的场所由开发者指导完成的测试。开发者记录错误和问题。

（4）β测试：是一个或多个用户的现场由该软件的最终用户实施的，开发者不在现场，拥护者记录错误和问题，并反馈给开发者。

（5）回归测试：是测试软件变更之后，变更部分的正确性和需求的符合性。只要软件变更，就要进行相应的回归测试。

在软件系统中的功能将模块分为四种类型。①传入模块:取得数据或输入数据，经过某些处理，再将其传送给其他模块.②传出模块：输出数据，在输出 之前可能进行某些处理，数据可能被输出到系统的外部，或者会输出到其他模块进行进一步处理。③变换模块：从上级调用模块得到数据，进行特定的处理，转换成其他形式，在将加工结果返回给调用模块。④协调模块一般不对数据进行加工，主要是通过调用、协调和管理其他模块来完成特定的功能。

需求分析确定软件要完成的功能及非功能性要求；概要设计将需求转化为软件的模块划分，确定模块之间的调用关系；详细设计将模块进行细化，得到详细的数据结构和算法；编码根据详细设计进行代码的编写，得到可以运行的软件，并进行单元测试。

软件设计的任务是基于需求分析的结果建立各种设计模型，给出问题的解决方案。从工程管理的角度，可以将软件设计分为两个阶段：概要设计阶段和详细设计阶段。结构化设计方法中，概要设计阶段进行软件体系结构的设计、数据设计和接口设计；详细设计阶段进行数据结构和算法的设计。面向对象设计方法中，概要设计阶段进行体系结构设计、初步的类设计/数据设计、结构设计；详细设计阶段进行构件设计。  
结构化设计和面向对象设计是两种不同的设计方法，结构化设计根据系统的数据流图进行设计，模块体现为函数、过程及子程序；面向对象设计基于面向对象的基本概念进行，模块体现为类、对象和构件等。

1999年ITU批准了五个IMT-2000的无线电接口，这五个标准是：  
IMT-DS(Direct Spread)：即W-CDMA,属于频分双工模式，在日本和欧洲制定的UMTS系统中使用。  
CDMA-2000,属于频分双工模式，是第二代CDMA系统的继承者。  
TD-SCDMA，属于时分双工模式。  
EDGE,是一种2.75G技术。  
IMT-2000 OFDMATDDWMAN，即无线城域网技术。

ADSL Modem上网拨号方式有3种，即专线方式(静态IP)、PPPoA和PPPoE。  
PPPoE(英语：Point-to-Point Protocol Over Ethernet)，以太网上的点对点协议，是将点对点协议(PPP)封装在以太网(Ethernet)框架中的一种网络隧道协议。  
PPTP(Point to Point Tunneling Protocol)，即点对点隧道协议。该协议是在PPP协议的基础上开发的一种新的增强型安全协议，支持多协议虚拟专用网(VPN)，可以通过密码验证协议(PAP)、可扩展认证协议(EAP)等方法增强安全性。可以使远程用户通过拨入ISP、通过直接连接Internet或其他网络安全地访问企业网。  
SLIP(Serial Line Internet Protocol，串行线路网际协议)，该协议是Windows远程访问的一种旧工业标准，主要在Unix远程访问服务器中使用，现今仍然用于连接某些ISP。  
PPP(点到点协议)是为在同等单元之间传输数据包这样的简单链路设计的链路层协议。这种链路提供全双工操作，并按照顺序传递数据包。设计目的主要是用来通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机、网桥和路由器之间简单连接的一种共通的解决方案。

立即寻址。操作数就包含在指令中。  
•直接寻址。操作数存放在内存单元中，指令中直接给出操作数所在存储单元的地址。  
•寄存器寻址。操作数存放在某一寄存器中，指令中给出存放操作数的寄存器名。  
•寄存器间接寻址。操作数存放在内存单元中，操作数所在存储单元的地址在某个寄存器中。  
•间接寻址。指令中给出操作数地址的地址。  
•相对寻址。指令地址码给出的是一个偏移量(可正可负)，操作数地址等于本条 指令的地址加上该偏移量。  
•变址寻址。操作数地址等于变址寄存器的内容加偏移量。

敏捷方法中，重构是一种重新组织技术，重新审视需求和设计，重新明确地描述它们以符合新的和现有的需求，可以简化构件的设计而无需改变其功能或行为。

RUP对软件开发过程的描述。RUP应用了角色、活动、制品和工作流4种重要的模型元素，其中角色表述“谁做”，制品表述“做什么”，活动表述“怎么做”，工作流表述“什么时候做”。

模块结构图的主要组成有：模块、调用、数据、控制信息和转接符号。

仓库风格是一种软件体系结构，其中包含一个数据仓库和若干个其他构件。数据仓库位于该体系结构的中心，其他构件访问该数据仓库并对其中的数据进行增、删、改等操作。数据库系统、超文本系统和黑板系统都属于仓库风格。  
该体系结构的优点包括：  
①对可更改性和可维护性的支持；  
②可复用的知识源；  
③支持容错性和健壮性。  
缺点包括：  
①测试困难；  
②不能保证有好的解决方案；  
③难以建立好的控制策略；  
④低效；  
⑤昂贵的开发工作；  
⑥缺少对并行机制的支持。

结构化设计主要包括：  
①体系结构设计：定义软件的主要结构元素及其关系。  
②数据设计：基于实体联系图确定软件涉及的文件系统的结构及数据库的表结构。  
③接口设计：描述用户界面，软件和其他硬件设备、其他软件系统及使用人员的外部接口，以及各种构件之间的内部接口。  
④过程设计：确定软件各个组成部分内的算法及内部数据结构，并选定某种过程的表达形式来描述各种算法。

数据字典是指对数据的数据项、数据结构、数据流、数据存储、处理逻辑、外部实体等进行定义和描述，其目的是对数据流程图中的各个元素做出详细的说明，使用数据字典为简单的建模项目。其条目有数据流、数据项、数据存储、基本加工等。

软件配置管理：

表更标识 变更控制（或变更管理） 版本控制 配置状态报告

（不包括风险管理 质量控制等）

统一过程（UP)：初启阶段、精化阶段、构建阶段、移交阶段和产生阶段。

初启阶段的里程碑是生命周期目标，

精化阶段的里程碑是生命周期架构，

构建阶段的里程碑是初始运作功能，

移交阶段的里程碑是产品发布。

封装 是一个对象把属性和行为封装为一个整体。

继承 是父类和子类之间共享数据和方法的机制

覆盖 子类在原有父类接口的基础上用适合于自己要求的实现去置换父类中的相应实现

多态 不同对象收到同一消息产生不同的效果。

●UML类图中类与类之间的关系有五种：依赖、关联、聚合、组合与继承。

若类A需要使用标准数学函数类库中提供的功能，那么类A与标准类库提供的类之间存在依赖关系；

（解释性语言没有编译，它是动态过程）

解释性：接收源程序，直接解释执行，读一句翻译一句执行一句，无目标代码。BASIC语言

编译性：将源程序直接翻译目标语言程序，包括编译全过程。C语言 C++语言。

编译程序不参与用户程序的运行控制，而解释程序则参与

总结为解释性程序无编译过程，无目标代码 编译性程序有编译过程，有目标代码。

脚本语言主要采用解释方式实现

编译的时候分配逻辑地址，运行的时候分配物理地址。

脚本语言都是动态语言，而动态语言都是解释型语言，

动态语言Php、Asp、JavaScript、Python、Perl

静态语言Java、C、C++、C#

中间代码可以使用三地址码 后缀式 树表示，而不能用堆和栈表示。

词法 输入字符，语法检查结构，语义（编译时报错为静态错误，运行时报错为动态错误。）

词法错误 非法字符标识符等

语法错误：单词拼写错误、标点符号错、表达式中缺少操作数、括号不匹配变量是否定义（或声明）等

语义阶段B.变量的值是否正确C.循环语句的执行次数 D.循环条件是否正确

逻辑错误，如死循环 分母为0等

词法分析 输入源程序，对源程序的字符串进行扫描分解，识别符号。有效工具为正规式和优先自动机。

语法分析 在词法分析的基础上，将单词符号分解为各类语法单位。如短语、句子、程序段

语法分析两种。（1）自上而下分析法（递归下降分析法和预测分析法）

（2）自下而上分析法（移进---归约分析法）

语义分析中间代码生成 对语法范畴进行静态语义检查，若正确则翻译为中间代码。通常使用属性文法描述语义规则。中间代码其实是含义明确便于处理的记号系统。

中间代码有三元式 间接三元式 四元式 树型 逆波兰记号。

常用的中间代码有后缀式、三地址码、树 。等。

优化 对中间代码加工，期望得到更有效的代码，优化是为等价交换过程。方法有公共子表达式的提取，循环优化 删除无用代码。

目标代码生成 把中间代码或优化后的代码转化为低级语言代码，依赖于机器指令/////////

采用面向对象的软件开发，通常有面向对象分析、面向对象设计、面向对象实现。

（1）面向对象分析是为了获得对应用问题的理解，其主要任务是抽取和整理用户需求并建立问题域精确模型。在面向对象分析阶段，并不考虑系统实现以及系统的测试问题，强调建立独立与系统实现的系统分析模型，

面向对象分析包含5个活动：认定对象、组织对象、描述对象间的相互作用、定义对象的操作、定义对象的内部信息。

（2）面向对象设计是采用协作的对象、对象的属性和方法说明软件解决方案的一种方式，强调的是定义软件对象和这些软件对象如何协作来满足需求，延续了面向对象分析。

（3）面向对象实现主要强调釆用面向对象程序设计语言实现系统。面向对象测试是根据规范说明来验证系统设计的正确性。

覆盖举例：

判定覆盖：（分支覆盖）整体真 整体假（2次/个）

条件覆盖：A真B假 A假B真

路径覆盖：所有路径覆盖

语句覆盖：所有语句覆盖

判定条件覆盖：A真B真 A假B假（2次/个）

系统开发计划：

PERT图 甘特图 工作分解表 预算分配表

人机交互“黄金三原则”包括：置于用户控制之下、减少用户的记忆负担、保持界面的一致性。

递归下降分析法和预测分析法是常用的自顶向下分析法。算符优先分析法和LR分析法属于自底向上的语法分析方法  
复合属性可以细分为更小的部分(即划分为别的属性)。  
学生的亲属的姓名可能有多个。这样的属性称为多值属性

程序的三种基本控制结构是顺序、选择和重复

计算机系统的存储器按所处的位置可分为内存和外存。

按构成存储器的材料可分为磁存储器、半导体存储器和光存储器。

按存储器的工作方式可分为读写存储器RAM和只读存储器ROM（断电丢失数据）。

RAM分为静态SRAM和动态DRAM SRAM就是cache，当前系统内存主要是DRAM

按访问方式可分为按地址访问的存储器和按内容访问的存储器。相联存储器是一种按内容访问的存储器。

按寻址方式可分为随机存储器、顺序存储器和直接存储器。

随机存储器 随机存储 顺序存取 存取时间与存储位置无关

顺序存储器 顺序存取 随机存储 如磁带 存取时间与存储位置有关

直接存储器 磁道随机 道内顺序 如磁盘 存取时间与存储位置有关

相联存储器 根据关键字与存储器中的每一单元对比查询，存取时间与存储位置无关

相联存储器是一种特殊的随机存储器。

全相联映射是指主存中任一块都可以映射到Cache中任一块的方式，也就是说，当主存中的一块需调入Cache时，可根据当时Cache的块占用或分配情况，选择一个块给主存块存储，所选的Cache块可以是Cache中的任意一块。  
直接相联映射方式是指主存的某块j只能映射到满足特定关系的Cache块i中。  
全相联映射和直接相联映射方式的优缺点正好相反，也就是说，对于全相联映射方式来说为优点的恰是直接相联映射方式的缺点，而对于全相联映射方式来说为缺点的恰是直接相联映射方式的优点。  
组相连映像了兼顾这两种方式的优点：主存和Cache按同样大小划分成块；主存和Cache按同样大小划分成组；主存容量是缓存容量的整数倍，将主存空间按缓冲区的大小分成区，主存中每一区的组数与缓存的组数相同；当主存的数据调入缓存时，主存与缓存的组号应相等，也就是各区中的某一块只能存入缓存的同组号的空间内，但组内各块地址之间则可以任意存放，即从主存的组到Cache的组之间采用直接映象方式；在两个对应的组内部采用全相联映象方式。

计算机病毒的分类

计算机病毒1引导型病毒 2文件型病毒 3混合型病毒（集两种与一体）

引导型病毒：改写磁盘上引导扇区内容，软盘硬盘皆可能感染。

文件型病毒：感染文件以文件名为COM EXE OVL等为主 它的安装必须借助于病毒载体程序

混合型病毒综合系统引导型病毒和文件型病毒特性。最难灭杀。

病毒种类的命名（核心）

（1）系统病毒：前缀为Win32 Win95 W32 W95 PE 感染Windows操作系统的.exe .dll文件

（2）蠕虫病毒：前缀为Worm 通过网络和系统漏洞传播，大部分蠕虫病毒向外发送带毒邮件，如冲击波（阻塞网络） 小邮差（带毒邮件）

（3）木马病毒：前缀Trojan 例如 Trojan.QQ3344 Trojan.Lmir.PSW.60

（4）黑客病毒：前缀Hake

（5）脚本病毒：前缀Script 通过网页传播，还有VBS JS脚本病毒如VBS.Happytime 等等

（6）宏病毒： 第一前缀Macro 第二前缀Word(DOC) Excel等等它是一种特殊的脚本病毒。

蠕虫病毒有红色代码、爱虫病毒、熊猫烧香、Nimda病毒、爱丽兹病毒等

（7）后门病毒：前缀Backdoor 该病毒通过网络传播，系统开后门，安全隐患大

（8）病毒种植程序病毒：体内释放新病毒，新病毒进行破坏，如冰河播种者（Dropper.Binghe2.2c）。MSN射手（Dropper.Worm.Smibag）

（9）破坏性程序病毒：前缀Harm 使用图标诱惑用户。如格式化C盘（Harm.formatC.f）

杀手命令（Harm.Command.Killer）

1. 玩笑病毒：前缀Joke 恶作剧病毒 如女鬼（Joke.Girl.ghost）
2. 捆绑机病毒：前缀Binder 将病毒与软件（QQ IE）等捆绑起来。如捆绑QQ（Binder.QQPass.QQBin） 系统杀手（Binder.killsys）等

/////////////////////////////////////////////////////////////////////

软件风险：

1软件风险包括不确定性和损失

2风险控制是在风险分析时，有效的辅助项目组处理风险。

3风险暴露包括风险概率及风险影响。

风险分析：风险识别，风险预测，风险评估，风险控制。

风险识别：试图系统化的确定对项目的威胁。

风险预测：风险发生可能性和后果。

风险评估：风险概率及后果是否影响参考水平值。

风险控制：有效的策略考虑风险避免，风险监控，风险管理，及意外事件。

软件成本估算模型：

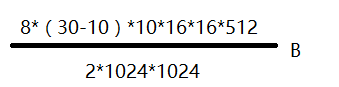
静态单变量模型：基本COCOMO

静态多变量模型：中级COCOMO

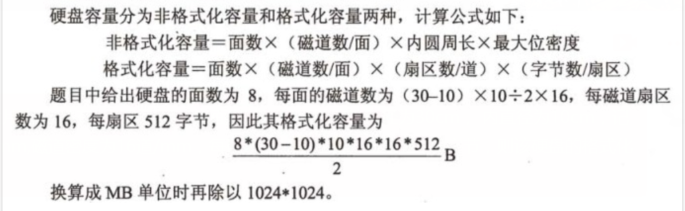
动态多变量模型：Putnam

存储管理

假设某硬盘由5个盘片构成(共有8个记录面)，盘面有效记录区域的外直径为30cm， 内直径为10cm，记录位密度为250位/mm,磁道密度为16道/mm，每磁道分16个扇区，每扇区512字节，

则该硬盘的格式化容量约为()MB。

则该硬盘的非格式化容量约为()MB。



A.CPU产生每条指令的操作信号并将操作信号送往相应的部件进行控制

C.CPU中的控制器决定计算机运行过程的自动化

D.指令译码器是CPU控制器中的部件

指令寄存器（IR)用来保存当前正在执行的指令。当执行一条指令时，先把它从内存取到数据寄存器（DR)中，然后再传送至IR指令寄存器

指令寄存器中操作码字段的输出就是指令译码器的输入，指令译码器：为了执行任何给定的指令，必须对操作码进行测试，以便识别所要求的操作

程序计数器PC的初始内容即是程序第一条指令的地址。并且执行一条指令，CPU中的值将自动加1

在程序控制方式下，由CPU执行程序控制数据的输入输出过程。  
在中断方式下，外设准备好输入数据或接收数据时向CPU发出中断请求信号，若CPU决定响应该请求，则暂停正在执行的任务，转而执行中断服务程序进行数据的输入输出处理，之后再回去执行原来被中断的任务。  
在DMA方式下，CPU只需向DMA控制器下达指令，让DMA控制器来处理数据的传送，数据传送完毕再把信息反馈给CPU,这样就很大程度上减轻了CPU的负担，可以大大节省系统资源。

其中MPEG-1（VCD DAB）、MPEG-2（DVD）和MPEG-4（Video）主要针对音、视频编码技术，而MPEG-7是多媒体内容描述接口标准，MPEG-21是多媒体应用框架标准。

软件配置管理主要包括变更标识、变更控制和版本控制。

软件变更控制是变更管理的重要内容，要有效进行变更控制，需要借助配置数据库和基线的概念。配置数据库一般包括开发库、受控库和产品库。不包括信息库

在传值方式下是将实参的值传递给形参，因此实参可以是表达式（或常量)，也可以是变量（或数组元素)，这种信息传递是单方向的在传地址方式下，需要将实参的地址传递给形参，因此，实参必须是变量（数组名或数组元素)必须有地址，不能是表达式（或常量)。 形参和实参间可以实现数据的双向传递

1. B.超大规模集成电路的简称 VLSI 超长指令字的简称 VLIW

C. 单指令流单数据流的简称 SISD 单指令流多数据流的简称 SIMD

多指令流单数据流的简称 MISD 多指令流多数据流的简称 MIMD

主存与Cache的地址映射方式中

全相联 随机将主存中的一块放入cache 但是很难寻找（例如随机停车，找不到车）

直接映射 按规定将主存的一块放入cache，难存放 但很容易寻找。

组相联 将主存分块，直接映射放入划分相等数目的cache中，在块内全相联映射。

（同时结合全相联和直接的优点。）

视频文件格式常用种类：AVI、mov、rmvb、rm、FLV、mp4、3GP

音频文件格式常用种类 CDA WAV MP3 WMA

图像文件格式常用种类 [BMP](https://www.baidu.com/s?wd=BMP%E6%A0%BC%E5%BC%8F&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank) GIF [JPEG](https://www.baidu.com/s?wd=JPEG%E6%A0%BC%E5%BC%8F&tn=SE_PcZhidaonwhc_ngpagmjz&rsv_dl=gh_pc_zhidao" \t "https://zhidao.baidu.com/question/_blank)

JPEG是一个适用范围很广的静态图像数据压缩标准

MPEG文件格式是运动图像压缩算法的国际标准

AVI格式允许视频和音频交错在一起同步播放

BMP以下不属于多媒体动态图像文件格式  
SWF属于多媒体动态图像文件格式

internet应用

ipconfig 显示信息；  
ipconfig /all 显示详细信息 ，可查看DHCP服务是否已启用；  
ipconfig /renew 更新所有适配器；  
ipconfig /release 释放所有匹配的连接。

ipconfig /flushdns 刷新和重置缓存的命令

在ASP的内置对象中，response对象和request 对象和cookie有关。其中，request对象中的Cookies集合是服务器根据用户的请求，发出的所有cookie的值的集合，这些Cookie仅对相应的域有效，每个成员均为只读。 Response对象中的Cookies集合是服务器发回客户端的所有Cookie的值，这个集合为只写，所以只有Response对象可以修改cookie中的值。

netstat用于显示与IP、TCP、UDP和ICMP协议相关的统计数据，一般用于检验本机各端口的网络连接情况。  
nslookup是一个监测网络中DNS服务器是否能正确实现域名的命令行工具。  
PING命令常用于测试连通性，在此过程中可看出是直接ping的目标地址。  
nslookup、ping、tracert都可以加上一个主机域名作为其命令参数来诊断DNS故障，nslookup还可以看到本地DNS服务器地址。Arp命令是与arp记录有关，与DNS无关联。

哈希表和二叉排序树都可以在查找过程中动态创建，属于动态查找表

C.二分查找要求査找表采用顺序存储结构或循环链表结构(此处错误)

D.顺序査找方法既适用于顺序存储结构，也适用于链表结构

A.连通无向网的最小生成树中，顶点数恰好比边数多1

C.可以采用AOV（使用AOE）网估算工程的工期

●下面关于栈和队列的叙述，错误的是D。

(60)A.栈和队列都是操作受限的线性表

B.队列采用单循环链表存储时，只需设置队尾指针就可使入队和出队操作的时间复杂度都为O(1)

C.若队列的数据规模n可以确定，则采用顺序存储结构比链式存储结构效率更高

D.利用两个栈可以模拟一个队列的操作，反之亦可（此处错误）

A.完全二叉树的髙度h与其节点数n之间存在确定的关系

邻接矩阵和邻接表是图（网）的两种基本存储结构，对于具有n个顶点、e条边的图， A.进行深度优先遍历运算所消耗的时间与采用哪一种存储结构无关

B.进行广度优先遍历运算所消耗的时间与采用哪一种存储结构无关

C.采用邻接表表示图时，查找所有顶点的邻接顶点的时间复杂度为O(n\*e)

优先队列通常采用堆数据结构实现。

队列插入元素时间复杂度O（lgn）

对于线性表 访问时间复杂度O（1） 插入时间复杂度O（n）

对于链表结尾，插入和删除的时间复杂度分别为O（1），O（n）

采用顺序表和单链表进行序号查找时间复杂度分别为O（1），O（n）

对于线性表，相对于顺序存储采用链表存储的缺点：数据元素不占存储空间，存储密度低。

顺序存储平均移动（n-1）/2；单链表存储平均移动为0；

对于栈和队列，若用单链表表示，则入队出队，入栈 出栈均不需要遍历链表。

C.完全图适合采用邻接矩阵存储

http 为协议类型  
IMG_256edu.abc.com 为域名，其中 edu 为主机名（服务器名称）  
 /lecturer/index/user\_id-8371666.html 为文件路径及文件名  
  
若用顺序表存储，则最好情况是删除最后一个元素，此时不用移动任何元素，直接删除，最差的情况是删除第一个元素，此时需要移动n-1个元素，所以平均状态是移动(n-1)/2。  
若用链表存储，直接将需要删除元素的前趋next指针指向后继元素即可，不需要移动元素，所以移动元素个数为0。  
  
生成器模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。  
实用范围  
1 当创建复杂对象的算法应该独立于该对象的组成部分以及它们的装配方式时。  
2 当构造过程必须允许被构造的对象有不同表示时

数据流图

题型1 根据数据流图确定里面内容。一般顶层图缺失对象，0层图缺失某某表或某某文件

题型2 缺失数据流，要求补充数据流图。

1. 先找是否有完全没有连线的。
2. 其次找顶层图。
3. 最后是分析内容找关系。

题型3 分析补充加工。

1. 图形分析加工，两图对比，箭头表示分析，上面有方向和内容（数据流名称）
2. 内容分析加工，根据内容找出关键字（存在推理性）： 必须 合法等名词

题型4 理论题

例1：程序流程图适用场合和作用：

程序流程图通常在进行详细设计时使用，用来描述程序的逻辑结构。

例2：说明逻辑数据流图与物理数据流图的区别：

物理数据流图关注的是系统中的物理实体，物理数据流图用做系统构造和实现的技术性蓝图。

逻辑数据流图强调的是参与者所做的事情，逻辑数据流图用做系统分析的需求分析阶段起点。

例3：绘制下层图时注意的问题：

保持父图与子图的平衡，父图中某加工的输入输出数据流必须与子图的输入输出数据流在数量上和名字上相同。如果父图的一个输入或输出数据流对应子图中的几个输入或输出数据流，那么子图中组成这些输入或输出数据流的数据项全体也必然对应父图中某一数据流。父图与子图之间必然平衡。

数据库

题型1明确告诉补充几个联系及联系关系。

方法：找到实体与实体的联系动词或名词，然后在找联系类型。

联系名词例如：管理，所属，权限，负责，诊断。

题型2补充关系模式设计里面的空。

一般给文字或图，文字可以直接找，但是大多数不仅仅只是一个实体。

方法：先找以及可以看出来的，再找比较隐含的。分析必须要有的关键几个。

题型3设计关系模式中的主键与外键。

方法：主键一般为第一个（或者最重要的）

外键就是自己有的，别的作为主键。（容易考负责人 经理等员工）

题型4新增功能满足实体联系关系。注意关系类型的分析

例如1 医生 药品 病人等（处方为联系）实体关系图。以及关系模式

药品（题目等具体内容。）

例如2 经理 权限 员工（设置 修改为联系）实体关系图。以及关系模式。

权限（员工号 权限 设置权限人等等）

题型5数据库理论  
例1 说明某关系模式的问题及解决方式。

某关系模式存在数据冗余，（或插入异常，修改异常，删除异常等操作），及数据修改的不一致性。可以考虑将表划分。

例如：若某关系模式有多个主键，则就存在此情况，要讲主键划分开。避免数据冗余。

例2 将某表合并的优劣比较。

优点：获得某些数据时，不必再进行连接查询，提高存储速度。

缺点：去掉某表之后，其内容会添加到合并表中，表中有大量数据，会造成数据冗余。同时若进行更新，则容易造成更新异常。甚至删除异常等操作。

UML

题型1用例图的关系

（1）包含：使用<<include>>表示 当从两个和两个以上的用例提取公共行为，使用包含关系，提取的公共用例称为抽象用例，而原始用例称为基本用例或基础用例。抽象用例必须是基础用例的一部分。

（2）扩展：使用<<extend>>表示 如果一个用例明显的混合两种或多种情景，即可能发生多种分支，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。抽象用例不一定是基础用例的一部分。

（3）泛化：当多个用例拥有一种类似的结构和行为的时候，可以将他们的共性抽象为父用例，其它关系作为泛化关系的子用例。

（核心）题型2 类图的类名与属性与多重度补充

类名：先把所有的跟英语有关的列出来，把类分一列，属性分一列。

重合度：（指向谁说明靠近谁，比如顾客类对行程类，一个顾客可以有多个行程选择）

则在靠近行程类旁边写1..\*（共有0..1 0..\* 1..1 1..\*）等等

C++

01 Public 考虚拟函数，要求根据子类函数写出虚拟函数

1. Virtual void/其它 +函数（）=0（2010下）

（2）类名 \*a=new 方法（void容易考调用）

protected（多）或private抽象函数与对象 （2010上）

例如创建对象：Flybehavior \*flybehavior

private（成员为公有函数调用）

若已经定义数组或者对象则调用即可 例（2014下）

若没有定义则this -->对象=对象。

02 Void函数：函数后考调用，类的调用使用：：（2013下）

对象的调用使用--->（指针）+将调用的函数（2009上）

调用括号没有内容就不写，

如果有考虑this或this---> 某某（对应函数已赋值给this指针）（2013上）

03 主函数 对象调用某个类中的函数是注意选择类（正确选择调用类也可用个数判断）

创建对象：类名 \*a=new 方法

04 聚合关系：聚合符号出口到入口，出口为聚合类，（写无符号一端）

例如聚合company类List<company \*> children(为对象可调用)（2010下）

company类List<company \*> \* children（）(为函数)（2009下）

里面的对象--->函数（属性）。可以没有属性。

例如（\*it）--->update（）（聚合类的方法）

05 组合关系的调用。组合符号出口到入口，入口为组合类，

例如组合调用 组合类调用子类。（2010上）

06 注意析构函数 ~函数名（）（2010上）

07 分类时注意对应的专属特征即可。（2009下）

在C++中继承的函数，在父类中使用virtual +函数：例如virtual void build（）

在Java中继承的函数，在父类中使用abstract+函数：例如abstract void build（）