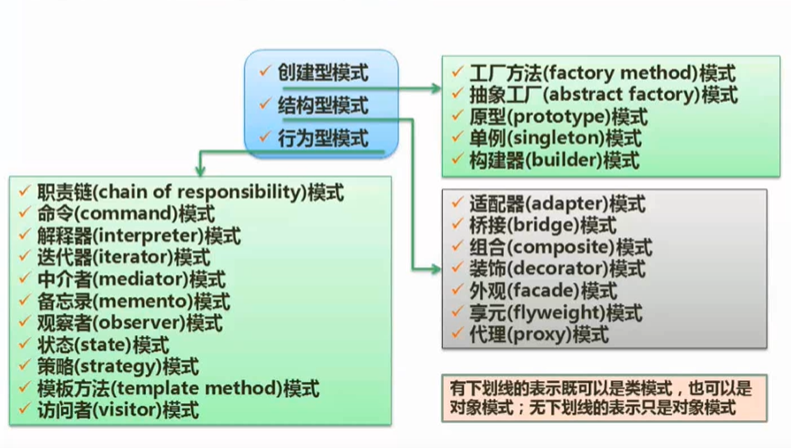
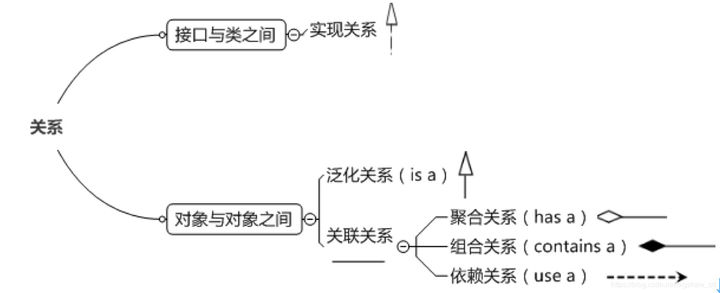
环路复杂度是封闭区域加1



UML图



词法分析：从左到右逐个扫描源程序中的字符，识别其中如关键字（或称保留字）、标识符、常数、运算符以及分隔符（标点符号和括号）等。

语法分析：根据语法规则将单词符号分解成各类语法单位，并分析源程序是否存在语法上的错误。包括：语言结构出错、if…end if不匹配，缺少分号、括号不匹配、表达式缺少操作数等。

语义分析：进行类型分析和检查，主要检测源程序是否存在静态语义错误。包括：运算符和运算类型不符合，如取余时用浮点数。

数据库体系结构：三层模式。对于题干给出的视图、存储文件、基本表分别对应：视图-外模式，存储文件-内模式，基本表-模式。

自然连接where左右都去除重复列

三种遍历方式：根左右(先序遍历)，左根右(中序遍历)， 左右根(后序遍历)

先序遍历第一个元素为根节点

常见的对称加密算法有：DES，三重DES、RC-5、IDEA、AES

公开密钥加密（public-key cryptography），也称为非对称加密（asymmetric cryptography），一种密码学算法类型，在这种密码学方法中，需要一对密钥，一个是私人密钥，另一个则是公开密钥。

常见的公钥加密算法有： RSA、ElGamal、背包算法、Rabin（RSA的特例）、迪菲－赫尔曼密钥交换协议中的公钥加密算法、椭圆曲线加密算法（Elliptic Curve Cryptography, ECC） ；DSA数字签名（又称公钥数字签名），将摘要信息用发送者的私钥加密，接收者只有用发送者的公钥才能解密被加密的摘要信息，也是属于公开密钥加密算法。

DES是典型的私钥加密体制，属于对称加密，不属于公开秘钥加密

以S结尾的是对称加密

DSA（Digital Signature Algorithm,数字签名算法）

ISO/IEC 9126软件质量模型，该模型的质量特性和质量子特性如下：

功能性（适合性、准确性、互用性、依从性、安全性）；

可靠性（成熟性、容错性、易恢复性）；

易使用性（易理解性、易学性、易操作性）；

效率（时间特性、资源特性）；

可维护性（易分析性、易改变性、稳定性、易测试性）；

可移植性（适应性、易安装性、一致性、易替换性）。

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。

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。公共的数据环境可以是全局数据结构、共享的通信区、内存的公共覆盖区等。

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。

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。

软件维护工具主要有：1、版本控制工具；2、文档分析工具；3、开发信息库工具；4、逆向工程工具；5、再工程工具；6、配置管理支持工具。

磁盘调度管理中，先进行移臂调度寻找磁道，再进行旋转调度寻找扇区。

codinger:

编译器的工作过程划分为词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成

词法分析的输出是记号流，也就是语法分析的输入

词法分析：

在词法分析阶段，其任务是从左到右逐个字符地读入源程序，对构成源程序的字符流进行扫描和分解，从而识别出一个个单词（也称单词符号或符号）。这里所谓的单词是指逻辑上紧密相连的一组字符，这些字符组合在一起才表示某一含义。词法分析过程依据的是语言的词法规则，即描述“单词” 分析构成程序的字符及由字符按照构造规则构成的符号是否符合程序语言的规定”是对单词的检查。

词法分析阶段依据语言的词法规则，对源程序进行逐个字符地扫描，从中识别出一个个“单词”符号，主要是针对词汇的检查。

从左到右逐个扫描源程序中的字符，识别其中如关键字（或称保留字）、标识符、常数、运算符以及分隔符（标点符号和括号）等。

词法分析的输出是记号流，也就是语法分析的输入

语法分析：

根据语法规则将单词符号分解成各类语法单位，并分析源程序是否存在语法上的错误。包括：语言结构出错、if…end if不匹配，缺少分号、括号不匹配、表达式缺少操作数等。括号不匹配是典型的语法错误

语法分析的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，如“表达式”“语句”和“程序”等。语法规则就是各类语法单位的构成规则，主要是针对结构的检查。

语义分析：

进行类型分析和检查，主要检测源程序是否存在静态语义错误。包括：运算符和运算类型不符合，如取余时用浮点数。

语义分析阶段分析各语法结构的含义，检查源程序是否包含语义错误、静态语言错误，并收集类型信息供后面的代码生成阶段使用。主要针对句子含义的检查。

### 案例分析题目类型

系统规划

软件架构设计

风格、架构设计与评估、质量属性、风险点、敏感点、权衡、非风险、ABSD概念、特点、阶段流程

设计模式

分类、概念、应用场景

系统设计

系统建模

逻辑建模：分析阶段 面向对象分析

物理建模：设计阶段 面向对象设计

分布式系统设计

嵌入式系统设计

系统可靠性分析与设计

冗余机制方法、计算可靠度

系统安全性和保密性设计

安全技术维度

UML

DFD、用例图、类图、顺序、状态、活动、协作

### 质量属性

六个非常常用的质量属性：可用性，可修改性，性能，安全性，可测试性，易用性。

**可用性**（Availability），可用性是指系统正常工作的时间所占的比例。可用性会遇到系统错误，恶意攻击，高负载等问题的影响。

**性能**，性能与时间有关。事件发生时，系统必须对其作出响应。时间到达响应有很多特性，但性能基本上于事件发生时，将要消耗系统多长时间做出响应有关系。

**安全性**是衡量系统在向合法用户提供服务的同时，阻止非法授权使用的能力。

**易用性**关注的是对用户来说完成某个期望任务的容易程度和系统所提供的用户支持的种类。

**可测试性**通过测试揭示软件缺陷的容易程度。

**可修改性**主要包含两方面，第一是修改什么（什么可以修改），第二个是何时以及由谁进行修改。

架构评估场景下，四大质量属性是：性能、安全性、可用性（在架构评估场景下不是可靠性）、可修改性

### 系统架构设计风格

需要重点掌握常见的软件设计风格

<https://www.jianshu.com/p/9a6699219f3a>

https://www.cnblogs.com/wintersun/p/4869344.html

### 设计模式

创建型模型：主要用于创建对象，为设计类实例化新对象提供指南。

结构型模式：主要用于处理类或对象组合，对类如何设计以形成更大的结构提供指南。

行为型模式：主要用于描述类或对象的交互以及责任的分配，对类之间交互以及分配责任的方式提供指南。

#### 设计模式的解决方案

一定要写定义一个接口，通过接口怎么怎么样。

例如：策略模式通过定义策略接口，通过策略接口实现类的方法，例如support方法，选择具体的实现类