

等级考试公共基础考点分析之软件工程基础(4) PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/180/2021_2022__E7_AD_89_E7_BA_A7_E8_80_83_E8_c97_180412.htm

考点16 设计准则 设计准则包括如下几点：(1)提高模块独立性；(2)模块规模适中；(3)深度、宽度、扇出和扇入适当；(4)使模块的作用域在该模块的控制域内；(5)应减少模块的接口和界面的复杂性；(6)设计成单入口、单出口的模块；(7)设计功能可预测的模块。

考点17 详细设计 详细设计主要确定每个模块具体执行过程，也称过程设计。详细设计的结果基本上决定了最终的程序代码的质量。详细设计的常用工具有：(1)程序流程图、N—PAD和HIPO；(2)判定表；(3)PDL(伪码)。

1程序流程图 程序流程图是一种传统的、应用广泛的、__巨有争议的软件设计表示工具。它用方框表示一个处理步骤，菱形代表一个逻辑条件，箭头表示控制流。

程序流程图的5种控制结构 (1)顺序型；(2)选择型；(3)先判断重复型；(4)后判断重复型；(5)多分支选择型。

程序流程图的主要缺点如下：(1)程序流程图从本质上不支持逐步求精，它诱使程序员过早地考虑程序的控制流程，不去考虑程序的全局结构；(2)程序流程图中用箭头代表控制流，使得程序员不受任何约束，可以完全不顾结构化设计的原则，随意转移控制；(3)程序流程图不易表示数据结构；(4)程序流程图的每个符号对应于源程序的一行代码，对于提高大型系统的可理解性作用甚微。

2N-S图 为了避免流程图在描述程序逻辑时的随意性与灵活性，1973年Nossi和Shneiderman提出了用方框图代替传统的程序流程图，引起了人们的重视，人们也把这种图称为N-S图。方框图中仅含5

种基本的控制结构，即顺序型、选择型、多分支选择型、WHILE重复型和UNTIL重复型。在方框图中，每个处理步骤都是用一个盒子来表示的，这些处理步骤可以是语句或语句序列，在需要时，盒子中还可以嵌套另一个盒子，嵌套深度一般没有限制，只要整张图可以在一张纸上容纳下就行。

N-S图有以下特点：(1)每个构件具有明确的功能域；(2)控制转移必须遵守结构化设计要求；(3)易于确定局部数据和(或)全局数据的作用域；(4)易于表达嵌套关系和模块的层次结构。

3 PAD图 PAD是问题分析图(Problem Analysis Diagram)的英文缩写。它是继流程图和方框图之后，由日本的二村良彦等人在1979年提出的又一种主要用于描述软件详细设计的图形表示下具。与方框图一样，PAD也只能描述结构化程序允许的几种基本结构、PAD图的一个独特之处在于，以PAD为基础，遵循一个机械的规则就能方便地编写出程序，这个规则称为走树(free Walk)。PAD图的基本图符表示5种基本控制结构 PAD图有以下特征：(1)结构清晰，结构化程度高；(2)易于阅读；(3)最左端的纵线是程序主干线，对应程序的第一层结构；每增加一层PAD图向右扩展一条纵线，故程序的纵线数等于程序的层数；(4)程序执行从PAD图最左主干线上端结点开始，自上而下、自左向右依次执行，程序终止于最左主干线。

4 PDL PDL又称伪码(Pseudo-code)，它是一种非形式化的比较灵活的语一言，实际上PDL语言是对伪码的一种补充、它借助于某些高级程序语一言的控制结构和一些自然语言的嵌套。一般说来，伪码的语法规则分成外语法和内语法外语法应当符合一般程序设计语言常用的程序语句的语法规则；而内语法是没有定义的，它可以用自然语言的一

些简洁的句子、短语和通用的数学符号来描述程序应该执行的功能。用PDL表示的基本控制结构的常用词汇如下：顺序：A/A END 条件：IF/THEN/ELSE/ENDIF 循环：DO

WHILE/ENDDO 循环：REPEAT UNTIL/ENDREPEAT 分支：CASE OF , /WHEN/SELECT/WHEN/SELECT/ENDCASE

PDL具备以下特征：(1)有为结构化构成元素、数据说明和模块化特征提供的关键词语法；(2)处理部分的描述采用自然语言语法；(3)可以说明简单和复杂的数据结构；(4)支持各种接口描述的子程序定义和调用技术。

3.4 软件的测试 考点18 软件测试的目的和准则 1软件测试的目的 (1)软件测试是为了发现错误而执行程序的过程；(2)一个好的测试用例能够发现至今尚未发现的错误；(3)一个成功的测试是发现了至今尚未发现的错误。

2软件测试的准则 (1)所有测试都应追溯到需求；(2)严格执行测试计划，排除测试的随意性；(3)充分注意测试中的群集现象；(4)程序员应避免检查自己的程序；(5)穷举测试不可能；(6)妥善保存测试计划、测试用例、出错统计和最终分析报告，为维护提供方便。

考点19 软件测试技术与方法 在软件生命周期的各个阶段，都有可能产生差错，虽然在每个阶段结束之前都有严格的复审，以期能尽早地发现错误。但是，经验表明审查并不能发现所有差错。如果在软件投入生产性运行之前，没有发现纠正软件中的大部分错误，则这些错误迟早会在运行过程中暴露出来，甚至会造成严重的后果。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com