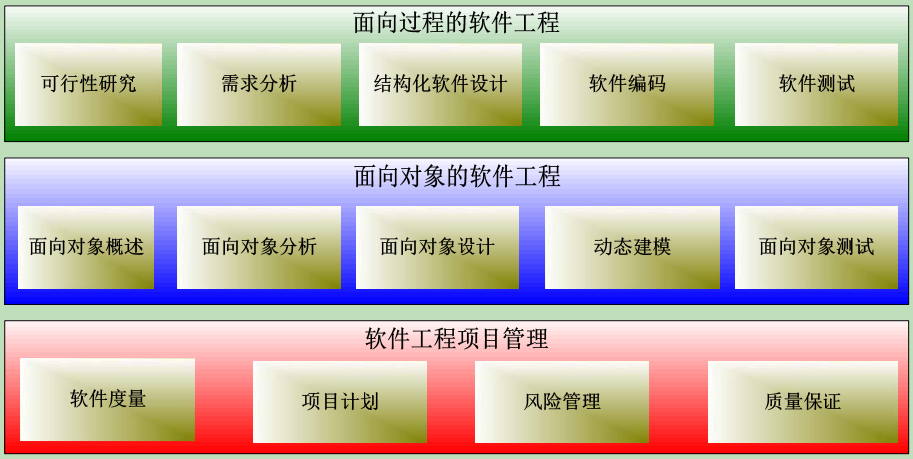
软件工程
第6讲 软件编码

贾西平
Email: jiaxp@126.com

课程主要内容



引言

- **编码阶段**的**任务**是根据详细设计说明书编写程序
- 程序设计语言的特性和程序设计风格会深刻**影响软件的质量和可维护性**
- 程序员必须深刻理解、熟练掌握并正确地运用程序设计语言的特性
- 尽可能使源程序有**良好的结构性**和**良好的程序设计风格**

08:46 3

提纲

- 程序设计语言
- 编程风格及软件效率
- 程序复杂度

08:46 4

提纲

- **程序设计语言**
- 编程风格及软件效率
- 程序复杂度

程序设计语言

- 分类
- 特性
- 选择方法

程序设计语言分类

- **按语言级别：面向机器语言和高级语言；**
- **按应用范围：通用语言和专用语言；**

08:46 7

面向机器语言

- 包括机器语言和汇编语言两种
  - 机器语言是计算机系统可以直接识别的程序设计语言。
  - 汇编语言是一种符号语言，采用了一定的助记符来替代机器语言中的指令和数据。
    - 汇编语言程序必须通过汇编系统翻译成机器语言程序，才能在计算机上运行。

08:46 8

高级语言

- 语句标识符与人类的自然语言较为接近，采用十进制数据表示形式，利于学习和掌握
- 抽象级别较高，不依赖于实现它的计算机硬件，编码效率较高
- 高级语言程序需经过编译或解释之后，才能生成可在计算机上执行的机器语言程序

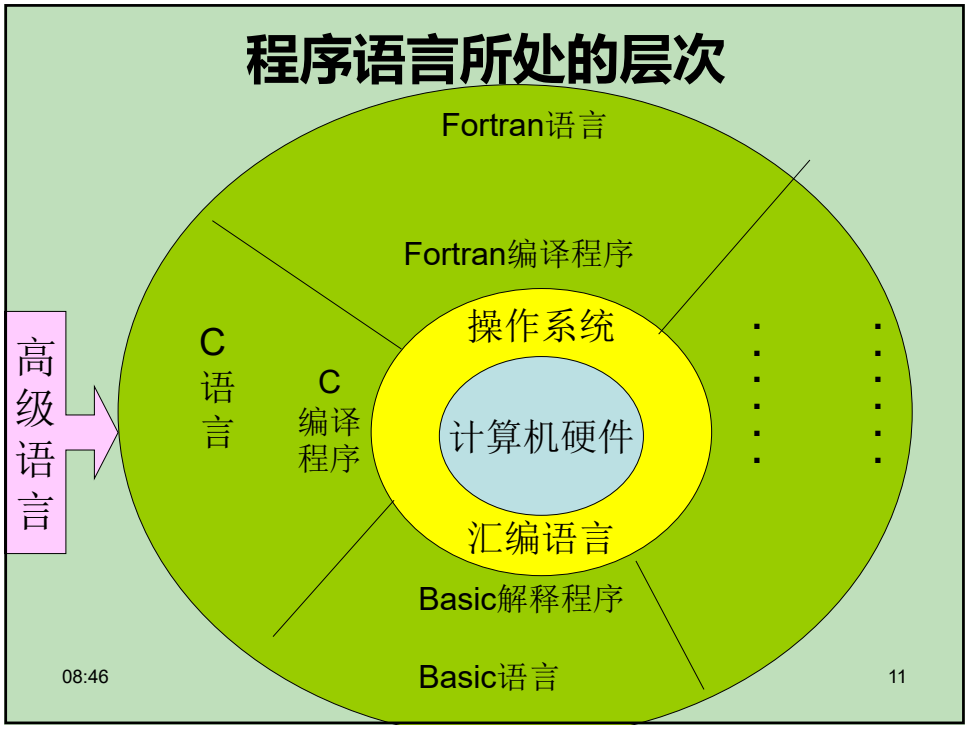
08:46 9

高级语言分类

按应用特点分类：通用语言和专用语言

- **通用语言**：可用于解决各类问题、广泛应用于各个领域的程序设计语言。
  - 如Basic、FORTRAN、C、Visual C、Java等
- **专用语言**：为了解决某类特殊领域的问题而专门设计的具有独特语法形式的程序设计语言。如：
  - APL：专用于解决数组和向量计算问题；
  - BLISS：专用于开发编译程序和操作系统程序；
  - LISP和PROLOG：专用于处理人工智能领域问题
  - **专用语言共同点**：可高效地解决本领域的各种问题，但难以应用于其他领域。

08:46 10



程序设计语言的特性

主要有三大特性

- **心理特性**
- **工程特性**
- **技术特性**

08:46 12

心理特性

- **程序设计语言的心理特性**
  能够影响编程者心理的语言性能。
- 主要有：
  - 歧义性
  - 简洁性
  - 局部性和顺序性

08:46 13

心理特性1：歧义性

- **歧义性：**程序设计语言中的某些语法形式使不同的人产生不同的理解。
  - 如FORTRAN语言中的表达式x**y**z，有人理解为(x**y)**z，有人却理解为x**(y**z)。

08:46 14

心理特性2：简洁性

- **简洁性**：编程者要使用该语言所必须记住的有关各种语法规则的信息量。如：
  - 语句格式
  - 数据类型
  - 运算符
- 好的程序设计语言，既应有一定的简洁性，又要有较高的可理解性

08:46 15

心理特性3：局部性和顺序性

- **局部性**：语言的联想性，即相关内容的相对集中性
  - 要求模块高内聚、低耦合，就是希望加强程序的局部性
- **顺序性**：语言的线性特征。
  - 例如，顺序结构的程序很容易理解，存在大量分支结构和循环结构的程序相对不易理解。
- 局部性和顺序性的加强可提高程序的可理解性。

08:46 16

工程特性

- **工程特性：**从软件工程的观点考虑为了满足软件开发项目的需要，程序设计语言应具备的特性。
- 主要包括：
  - 可移植性
  - 语言编译器的实现效率
  - 开发工具的支持
  - 可维护性

工程特性1：可移植性

- 程序在不同机器环境下的通用性和适应性。

工程特性2：语言编译器的实现效率

- 不同语言的编译器生成目标代码的大小和执行效率不尽相同
- 为获得高效的目标代码，选择语言时应充分考虑语言编译器的实现效率

工程特性3：开发工具的支持

- 优先选择具有良好开发工具支持的程序设计语言
- 开发工具主要包括：编译程序、连接程序、交互式调试器、交叉编译器等。

工程特性4：可维护性

- 采用的语言应具有良好的可读性和易于使用，以提高程序的可维护性

技术特性

- 语言的技术特性
  - 丰富的数据类型
  - 复杂的数据结构
  - 较强的实时处理能力
  - …
- 根据项目的特性选择具有相应技术特性的程序设计语言

程序设计语言的选择

- 综合考虑各种实际因素
- 兼顾**理论标准**和**实用标准**

08:46 23

理论标准

1. 理想的模块化机制、易于阅读和使用的控制结构及数据结构
2. 完善、独立的编译机制

08:46 24

实用标准

1) 系统用户的要求
2) 工程的规模
3) 软件的运行环境
4) 可以得到的软件开发工具
5) 软件开发人员的知识
6) 软件的可移植性要求
7) 软件的应用领域

提纲

- 程序设计语言
- **编程风格及软件效率**
- 程序复杂度

编码风格

- **编码风格**：在不影响程序正确性和效率的前提下，有效编排和合理组织程序的基本原则
- 编码风格主要体现在如下方面:
  - 内部文档
  - 标识符的命名及说明
  - 语句的构造及书写
  - 输入/输出

08:46 27

内部文档

- **内部文档**：程序中的注释信息
- 注释一般分两类：
  - **序言性注释**：一般位于模块的首部，用于说明模块的相关信息。
    - 如：模块的功能、用途、参数描述、需调用的下级模块清单、模块编写、审核、修改的相关信息等
  - **描述性注释**：位于源程序模块内部，用于对某些难以理解的语句段的功能或某些重要的标识符的用途等进行说明
    - 恰当的描述性注释可以大大提高程序的可读性和可理解性
    - 对语句的注释应紧跟在被说明语句之后

08:46 28

标识符的命名

标识符的命名应注意以下几点：

- 选用具有实际含义的标识符
  - 如：年龄变量命名为age
- 标识符的名字不宜过长
  - 通常不要超过八个字符，有些语言可以更多
- 不同的标识符不要取过于相似的名字
  - 如： student和students很容易混淆

标识符的说明

对其进行类型说明时应注意以下几点：

- 按某种顺序分别对各种类型的变量进行集中说明
  - 如：先说明简单类型，再说明指针类型，再说明记录类型。
- 使用一个说明语句对同一类型的多个变量进行说明时，应按照变量名中的字母顺序(a～z)进行排列

语句的构造

构造语句应注意问题：

(1)语句应简单直接，不用华而不实的程序设计技巧

(2) 对复杂表达式，加上必要的括号使表达更加清晰

(3)在条件表达式中尽量不使用否定的逻辑表示

(4) 尽量不使用强制转移语句GOTO

(5) 不要书写太复杂的条件，嵌套的重数不宜过多

(6) 尽可能地使用编译系统提供的标准函数

语句的书写

书写程序时需注意如下几点：

(1) 一行只书写一条语句

(2) 采用递缩式格式使程序的层次更清晰

(3) 在模块之间通过加入空行进行分隔

(4) 最好在注释段的周围加上边框

输入

输入应注意如下几点：

(1) 输入方式力求简单，避免给用户带来不必要的麻烦

(2) 交互式输入数据时应有必要的提示信息

– 提示信息可包括：输入请求、数据的格式及可选范围等。

(3) 程序应对输入数据的合法性进行检查

(4) 若用户输入某些数据后可能会产生严重后果，应给用户输出必要的提示，必要时要求用户确认。

(5) 需要输入一批数据时，应以特殊标记作为数据输入结束的标志。

(6) 根据系统的特点和用户的习惯设计出令用户满意的输入方式。

输出

设计数据输出方式时应注意以下几点：

(1) 输出数据的**格式应清晰、美观**。

(2) 输出数据时要加上**必要的提示信息**。

软件效率

用于提高运行速度的指导原则:

(1) 编写程序前，先**化简**要用的算术表达式和逻辑表达式。

(2) 尽可能**采用执行时间短的算术运算**。

(3) 尽量**避免使用复杂的数据类型**，如多维数组、指针等。

(4) 尽量**采用整型算术表达式**和**布尔表达式**

(5) 尽可能**减少循环体**，特别是**内循环中语句的个数**。

(6) 尽量**使同一表达式中的数据类型保持统一**。尽量**避免不同类型数据的比较运算**

(7) **对所有输入和输出安排适当的缓冲区**，以减少频繁通信所带来的额外开销

08:46 35

软件效率

用于优化存储空间使用的指导原则

(1) 对于变动频繁的数据最好**采用动态存储**。

(2) 按需**采用存储单元共享等节约空间的技术**。

(3) **选用具有紧缩存储器特性的编译程序**，在必要时甚至可采用汇编语言。

(4) **采用结构化程序设计，将程序划分为大小合适的模块**。

08:46 36

提纲

- 程序设计语言
- 编程风格及软件效率
- **程序复杂度**

软件质量评价

- **主要衡量标准**
  - **程序复杂度**
  - **可读性**
  - **效率**

程序复杂度度量方法

- 环形复杂度的度量
- 文本复杂度的度量
- 交点复杂度的度量

程序图

- **程序复杂度的主要研究对象**：程序结构的清晰性和非结构化程度
- 程序图**有助于表达程序结构**
- 程序图**只关心程序流程，不关心处理细节**
- 原来程序流程图中的各个**处理框**被简化为**结点**，用**圆圈**表示，程序流程图中带有箭头的**控制流**变成了程序图中的**有向边**

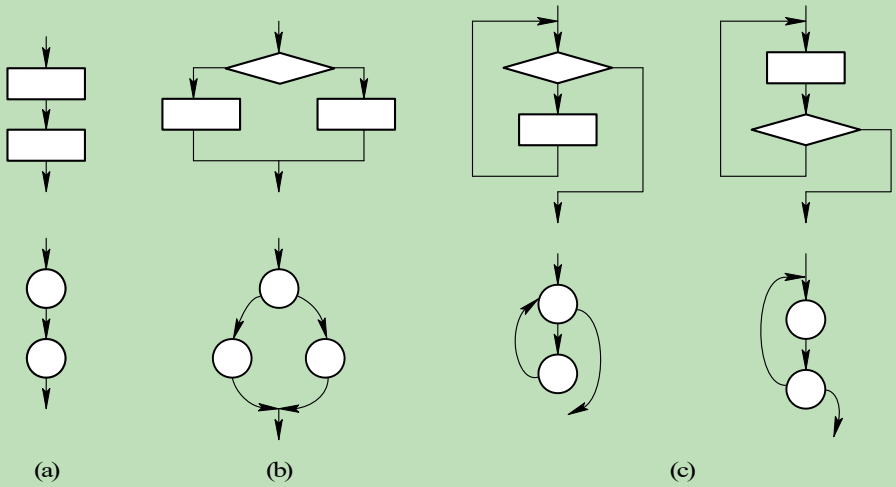


图5.1 几种基本结构的程序图
(a) 顺序结构；(b) 分支结构；(c) 循环结构(当型和直到型)

08:46 41

程序图

- 可通过简化程序流程图得到，也可以由PAD(Problem Analysis Diagram)图或其他详细设计表达工具变换获得。
- **入口点**：程序图开始点后面的那个结点。
- **出口点**：结束点前的那个结点。
- 结构化设计的程序通常只有一个入口点和一个出口点。

08:46 42

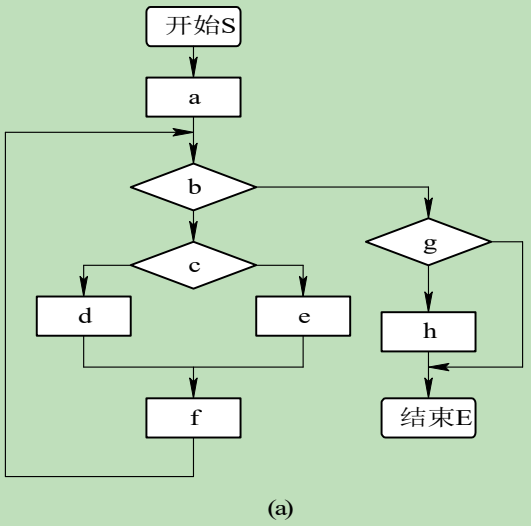


图5.2 程序流程图及导出的程序图
(a) 程序流程图；(b) 程序图

08:46 43

环形复杂度的度量

- **环形复杂度：**一个强连通的程序图中线性无关的有向环的个数。
- **强连通图**：从图中任意一个结点出发都能到达图中其他结点的有向图。
- **程序的环形复杂度的计算**：
  - 导出该程序的程序图
  - 若不是强连通图，增加一条从**结束点**到**开始点**的虚边
  - 用公式：

    V(G) = m - n + 1 ①

    V(G)--环形复杂度；**m**--有向图中有向边的个数；**n**--有向图中的结点个数；
  - 或公式：

    V(G) = 强连通的程序图在平面上围成的区域数 ②
  - 或公式：

    V(G) = 判定结点数 + 1 ③

08:46 44

方法1：
边：13，顶点：10
V(G)＝13－10＋1＝4

方法2：
围成的区域有：(b,c,d,f,b)，(c,d,f,e,c)，(g,h,E,g)和(S,a,b,g,E,S)，根据公式②可得该程序图的环形复杂度为4

方法3：
判定结点分别为b，c和g；
环形复杂度为：3＋1＝4

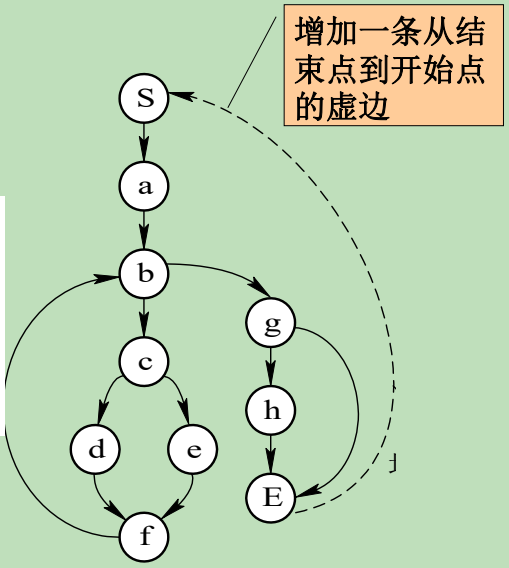


图5.3　增加虚边将原程序图转变为强连通图

08:46　45

程序的环形复杂度

- 研究发现，**程序的环形复杂度越高，可理解性就越差**，测试和维护的难度也越大
- **环形复杂度高的程序，容易出问题**
- 实践证明，**尽量将程序的环形复杂度控制在10以下**较好

08:46　46

文本复杂度的度量

- 文本复杂度的度量方法又称**Halstead**方法
- **原理**：根据源程序中**运算符**(包括关键字)和**操作数**(包括常量和变量)的总数来度量程序复杂度，可预测程序的**文本复杂度**和程序中的**错误数**
- 度量源程序文本复杂度的方法：
  - 找出整个程序中**运算符出现的总次数N1**及**操作数出现的总次数N2**
  - **程序的文本复杂度N = N1 + N2**

08:46 47

Halstead方法

- 详细设计结束后，可知程序中**不同运算符的个数n_1**和**不同操作数的个数n_2**
- **预测程序的文本复杂度H：**

$$H = n_1 \cdot \log_2 n_1 + n_2 \cdot \log_2 n_2$$

- **预测程序中包含的错误个数E：**

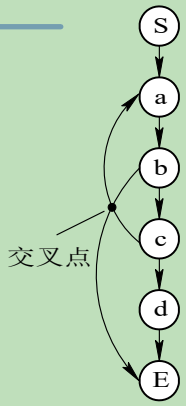
$$E = (n_1 + n_2) \log_2(n_1 + n_2)/3000$$

- 实践证明
  - 预测的程序文本复杂度H和实际程序的文本复杂度N十分接近
  - 预测出的程序错误数与实际错误数的相对误差不超过8%。

08:46 48

交点复杂度的度量

- **度量方法**：程序图中所有的转移线画在结点的**同一侧**，则**程序的交点复杂度**=程序图中交叉点的个数
  - 如图，有两条转移线相交于一点，因此该程序的交点复杂度为1。
- 严格采用结构化设计方法设计的程序中通常不含交叉点，交点复杂度为0。
- 若在程序中使用强制转移语句，就会增加程序的交点复杂度。
- 程序中应尽量**避免或减少GOTO语句**



08:46 49

要点回顾

- 程序设计语言的分类、特性、选择方法
- 编程风格的四个方面
  - 注释、标识符、语句构造、输入/输出
- 软件效率的改善原则
- 程序复杂度的度量：
  - 环形复杂度：
  ① V(G) = m - n + 1，
  ② V(G)=判定结点数 + 1
  ③ V(G) = 强连通的程序图在平面上围成的区域数
  - 文本复杂度
  ① N = N1 + N2
  ② $H = n_1 \cdot \log_2 n_1 + n_2 \cdot \log_2 n_2$
  ③ 错误数：$E = (n_1 + n_2) \log_2(n_1 + n_2)/3000$
  - 交点复杂度:同一侧、交叉点个数

08:46 50

思考题

思考下面问题：

- 程序设计语言有哪些特性？
- 程序设计语言有哪些选择标准？
- 编码风格体现在哪些方面？
- 程序复杂度的度量方法有哪些？如何计算？

08:46 51

视频资料

- 编程过程规范19’
  https://www.bilibili.com/video/BV1Q741157ve?p=7
- 良好编程实践18’
  https://www.bilibili.com/video/BV1Q741157ve?p=8
- 集成开发环境5’
  https://www.bilibili.com/video/BV1Q741157ve?p=9
- 代码静态检查13’
  https://www.bilibili.com/video/BV1Q741157ve?p=10
- 代码性能分析12’
  https://www.bilibili.com/video/BV1Q741157ve?p=11
- 结对编程32’(6’)
  https://www.bilibili.com/video/BV1Q741157ve?p=12
- 软件工程师经历分享16’
  https://www.bilibili.com/video/BV1Q741157ve?p=13

08:54 52

谢谢！

08:46 53