# 详细设计之任务与方法

当前软件产品开发过程中，由于需求的不断变更，需要开发团队采用更为快速敏捷的方法，即Rapid Prototype Modeling，但开发工作还是要一步步的来进行。

1.需求分析--产生软件功能规格说明书,需要确定用户对软件的需求,要作到明确、无歧义。不涉及具体实现方法。用户能看得明白，开发人员也可据此进行下面的工作（概要设计）。

2.概要设计--产生软件概要设计说明书，说明系统模块划分、选择的技术路线等，整体说明软件的实现思路。并且需要指出关键技术难点等。

3.详细设计--产生软件详细设计说明书，对概要设计的进一步细化，一般由各部分的担当人员依据概要设计分别完成，然后在集成，是具体的实现细节。理论上要求可以照此编码。

程序员需要详细的文档来告诉他们为什么要这样做，以及哪些不能做，把业务范围设定好。这些业务设定，必须用详细的图表+母语文字来描述。事实上，即使专业的产品团队在缺乏业务知识、专业知识的的情况下也很难设计出合理的产品。这时候就需要业务专家/领域专家来提供专业支持，同时需要技术人员来提供技术支持，告诉产品设计时底层技术和底层业务模型实现应该是是什么样的。

其实所谓的“设计”，就好像是写文章时打个草稿，目的是为了在实际写代码之前就把一些容易出问题的地方想好，避免编码时写错而付出更大的代价。但是设计本身也是有成本的，所以一般要根据业务本身的特性来做合适的设计流程。

如果跳过详细设计，直接编写代码的话，在软件规模小，业务简单的情况下是可行的。但对稍有复杂度的软件来说，缺乏详细设计的软件将来在面对需求变更、软件维护、人员离职时会遇到很大麻烦。但详细设计也没有必要特别精细，只要保证开发人员根据详细设计在以后实施中不会出大的理解偏差即可，因为越是设计的精细成本越高。

**一.详细设计的任务**

1.为每个模块确定采用的算法

2.确定每一模块使用的数据结构

3.确定模块接口的细节

4.要为灭一个模块设计出一组测试用例

**二.详细设计的方法**

在过程设计阶段，要决定各个模块的实现算法，并精确地表达这些算法。表达过程规格说明的工具叫做详细设计工具。它可以分为以下三类：

➽ 图形工具（程序流程图，N-S图，问题分析图，PAD）

➽ 表格工具 （判定表）判定路径是否正确

➽语言工具（program design language，PDL）

图形工具：

(一）程序流程图：也称程序框图

优点：易学、表达算法直观。

缺点：(1)不够规范，特别是使用箭头使质量受到很大影响，需要加以限制，使其成为规范的详细设计工具

(2)不是逐步求精的好工具，它诱使程序员过早考虑控制流程，而不去考虑程序的整体结构。

（3）不易表示数据结构和层次结构。

五种基本控制结构：

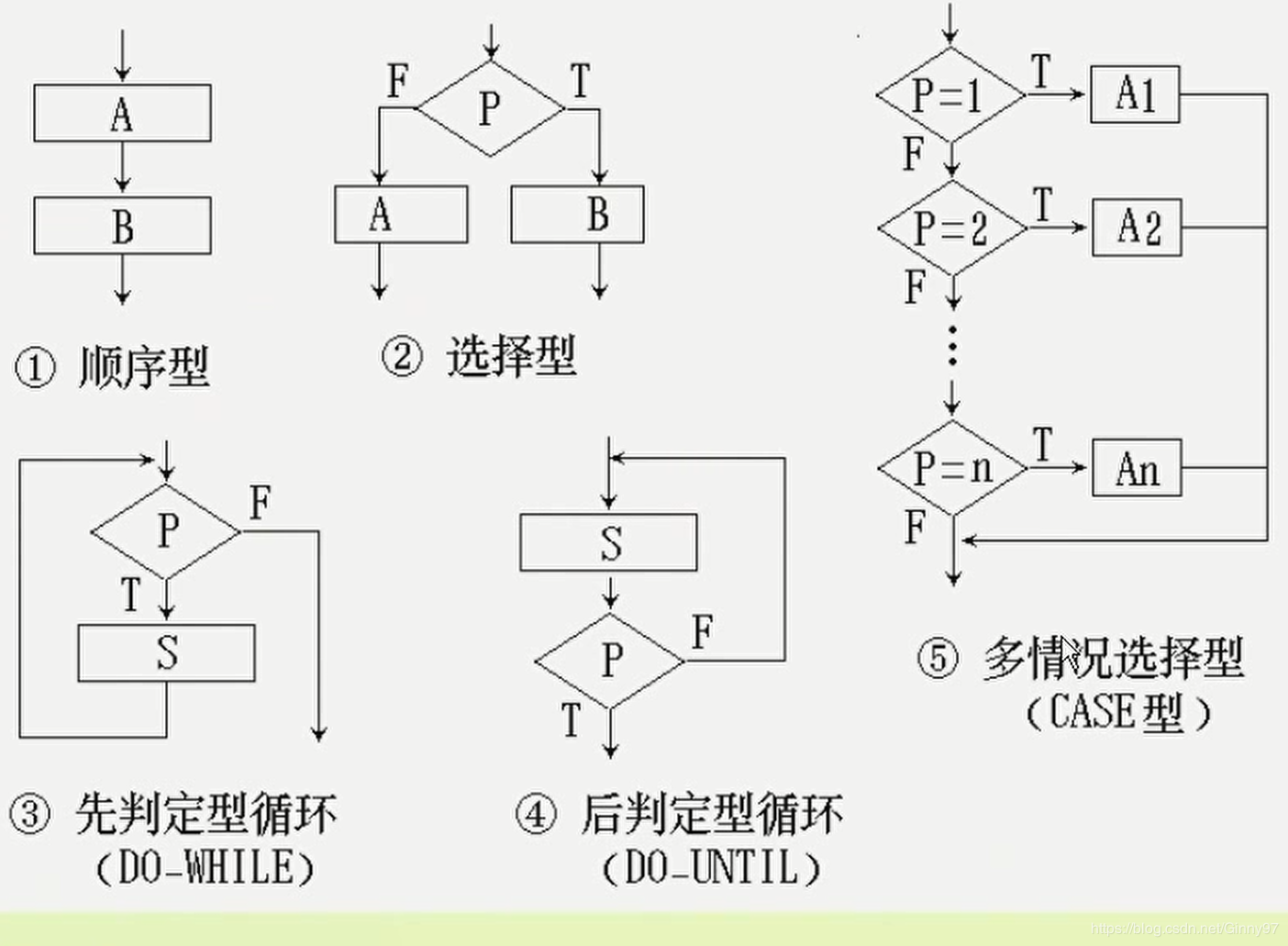
顺序型

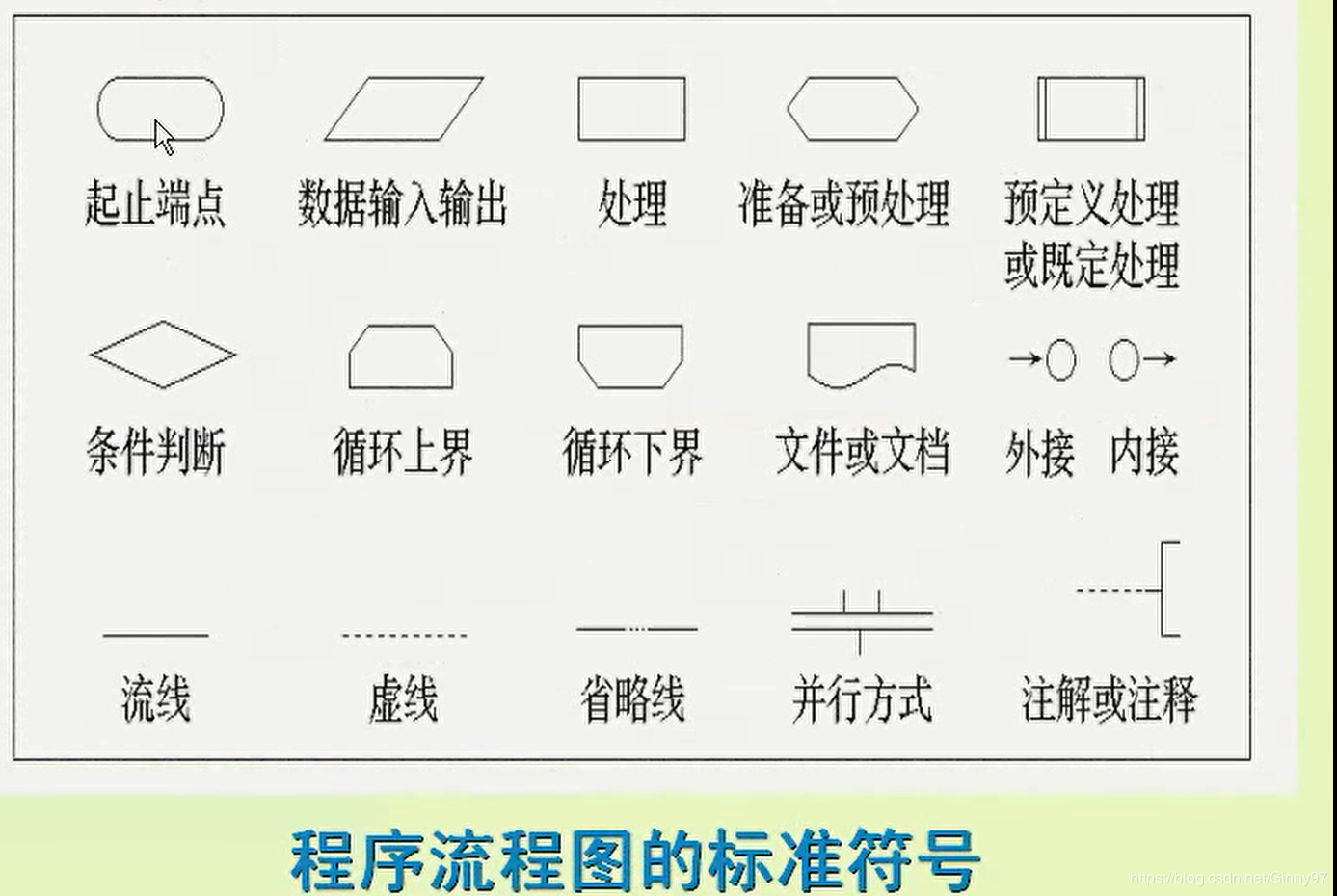
选择型

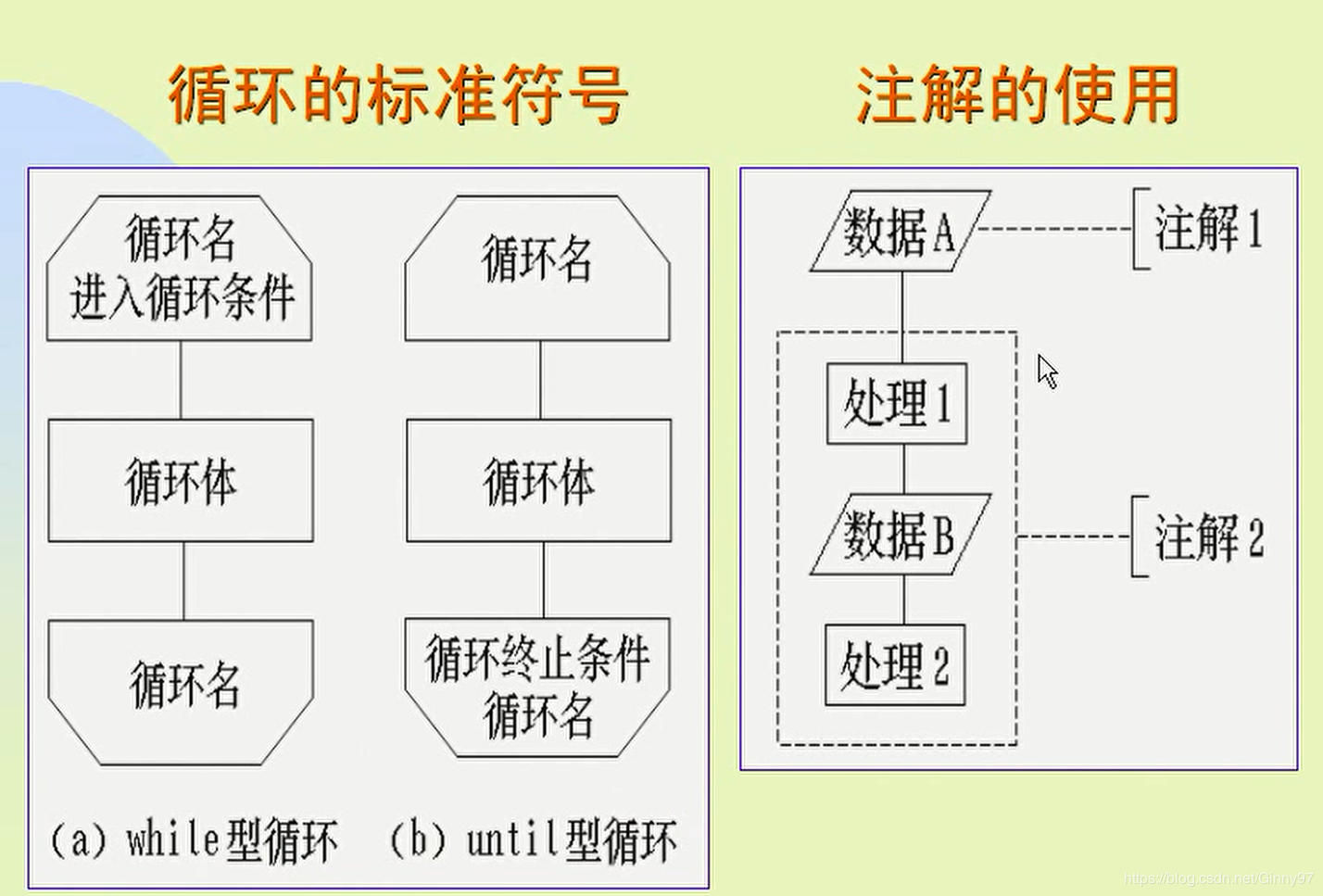
先判定型循环（Do-While)

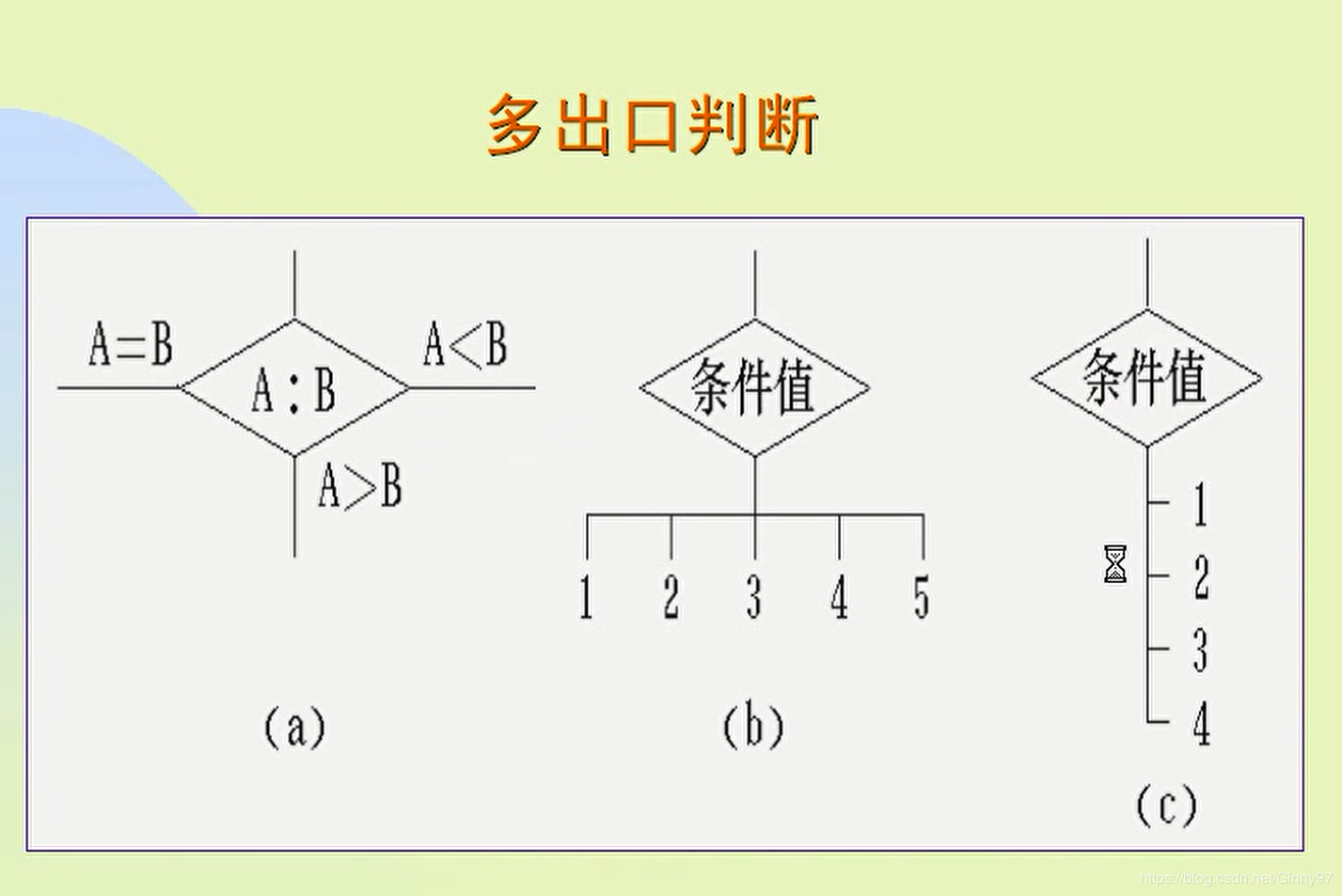
后判定型循环(Do-Until)

多情况选择型(CASE型）



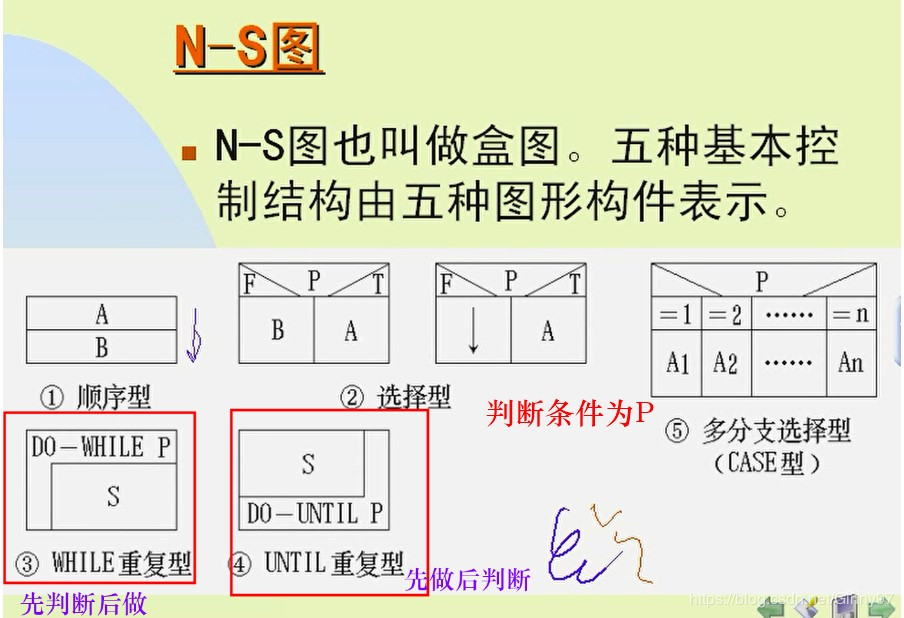






（二）N—S图（盒图）：

五种基本控制结构由五种图形构件表示：



N-S图特点：

1.功能域表示明确

2.不能实现任意的转移

3.很容易确定局部和全部的作用域

4.很容易的表示模块的嵌套以及模块的层次结构

5.可以表示模块之间的调用关系

N-S图的不足：

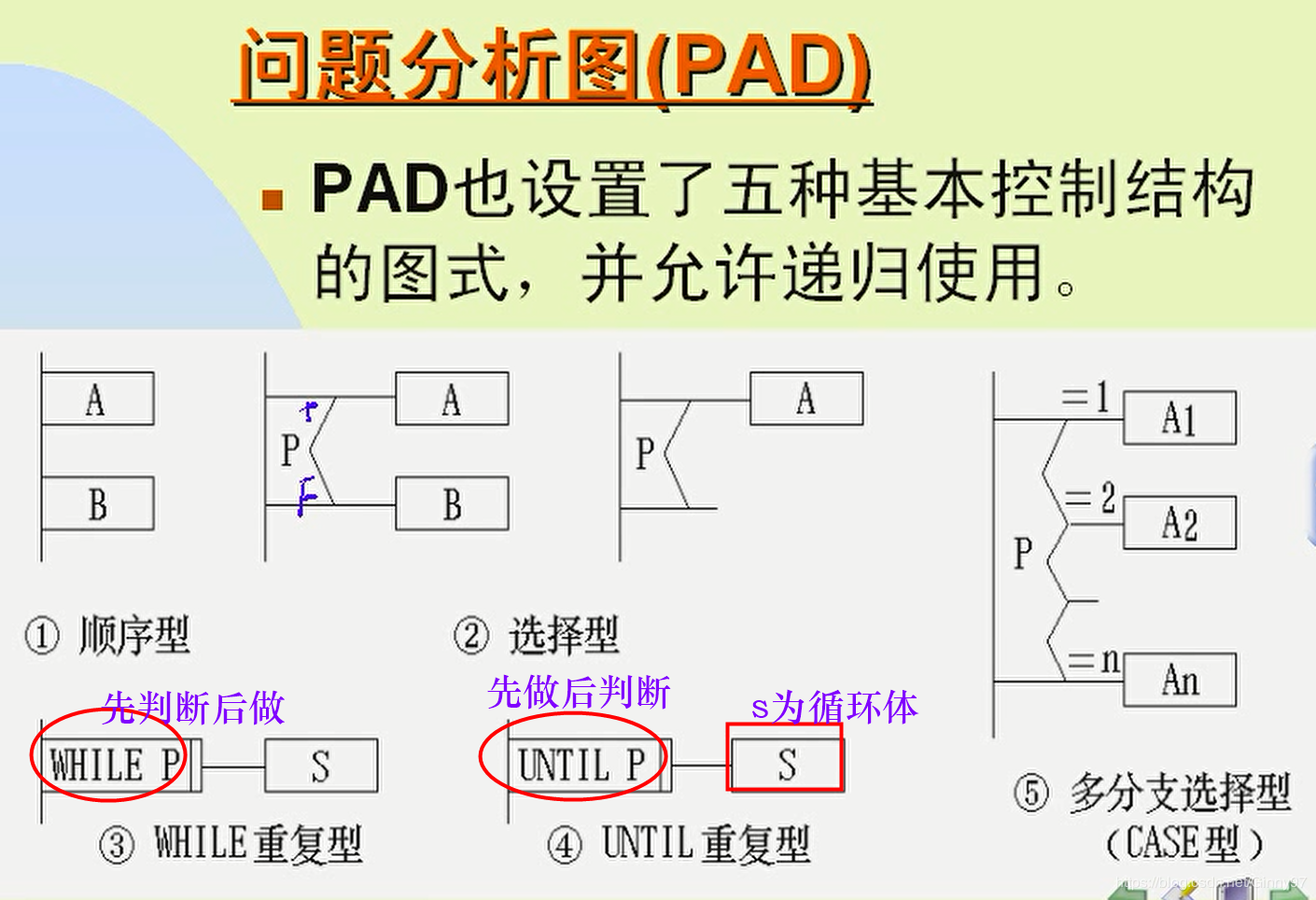
1.控制关系比较隐含

2.循环次数隐含

3.比程序流程图好 ，使编程人员逐步养成结构化方式思考问题

注意：N-S 图可以嵌套，但是嵌套不允许交叉。

（三）问题分析图（PAD）：



PAD图优点：

1.所设计的程序是结构化的

2.所描述的流程图清晰

3.程序的逻辑结构易懂，便于记忆

4.很易将PAD图转变为高级语言程序，嵌套层次清晰

5.很容易描述数据结构

6.支持自顶向下，逐步求精的方法

（四）HIPO图：HIPO 图

表格工具：

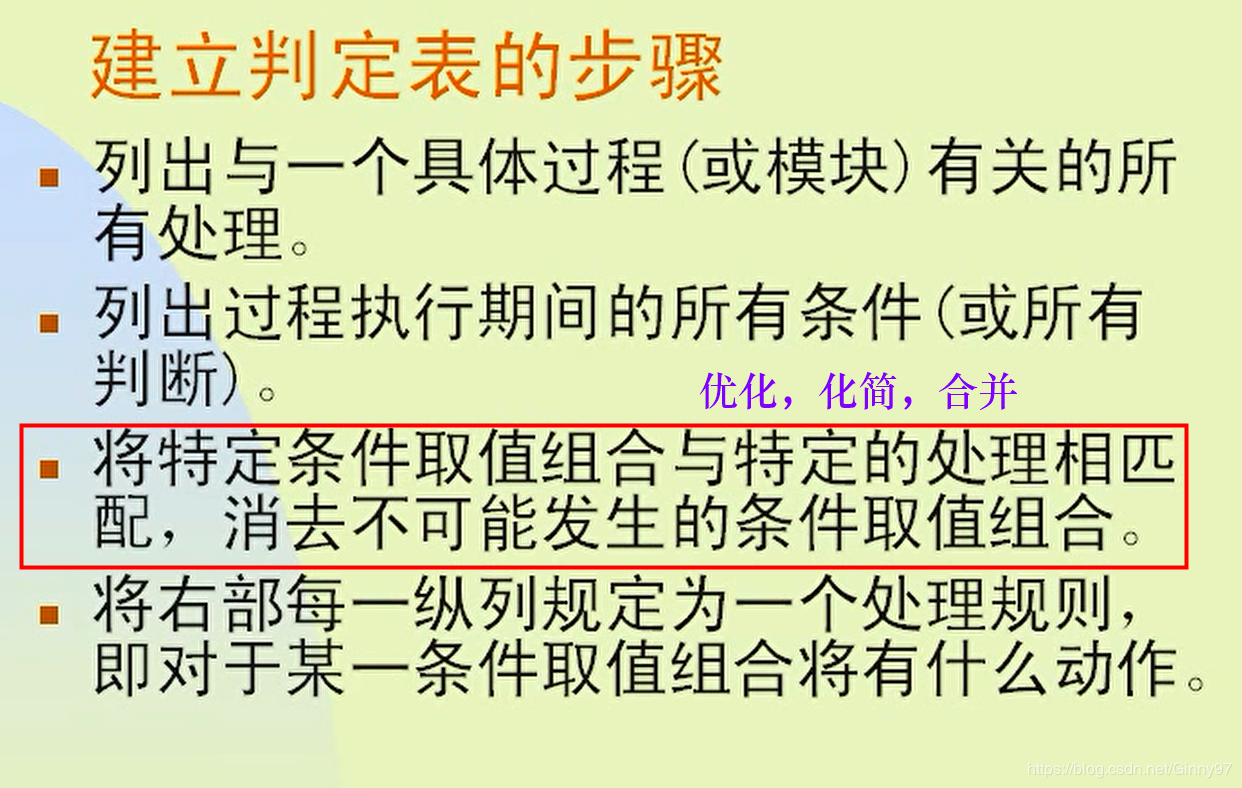
判定表：

❥ 判定表用于表示程序的静态逻辑

❥ 在判定表中的条件部分给出所有的两分支判断的列表，动作部分给出相应的处理

❥要求将程序流程图中的多分支判断都改成两分支判断

建立判定表的步骤：



语言工具：

过程设计语言（PDL）：

✦ 过程设计语言（Process Design Language, PDL）也称为伪码，是一种用于描述模块算法设计和处理细节的语言。

✦ 一方面，PDL具有严格的关键字外层语法，用于定义控制结构和数据结构;

✦另一方面，PDL表示实际操作和条件的内层语法又是灵活自由的，可使用自然语言的词汇。

PDL特点：

✎提供全部结构化控制结构、数据说明和模块特征。能对PDL正文进行结构分割，使之变得易于理解。

✎为了区别关键字，规定关键字一律大写，其他单词一律小写。

或者规定关键字加下划线，或者规定他们为黑体字。

PDL的优点：

✎它既可以描述过程，数据，也可以描述模块，接口等。

PDL不足：

✎不足：不如图形工具直观，复杂的条件组合与动作之间关系时不如判定表清晰简单。