

结构化组件设计

组件级设计也称为过程设计、详细设计，位于数据设计、体系结构设计和接口设计完成之后

任何程序总可以用三种结构化的构成元素来设计和实现

- 顺序：任何算法规约中的核心处理步骤
- 条件：允许根据逻辑情况选择处理的方式
- 重复：提供了循环

详细设计工具可以分为以下三类：

- 图形设计符号：流程图、盒图等
 - 表格设计符号：决策表等
 - 程序设计语言：PDL等
-

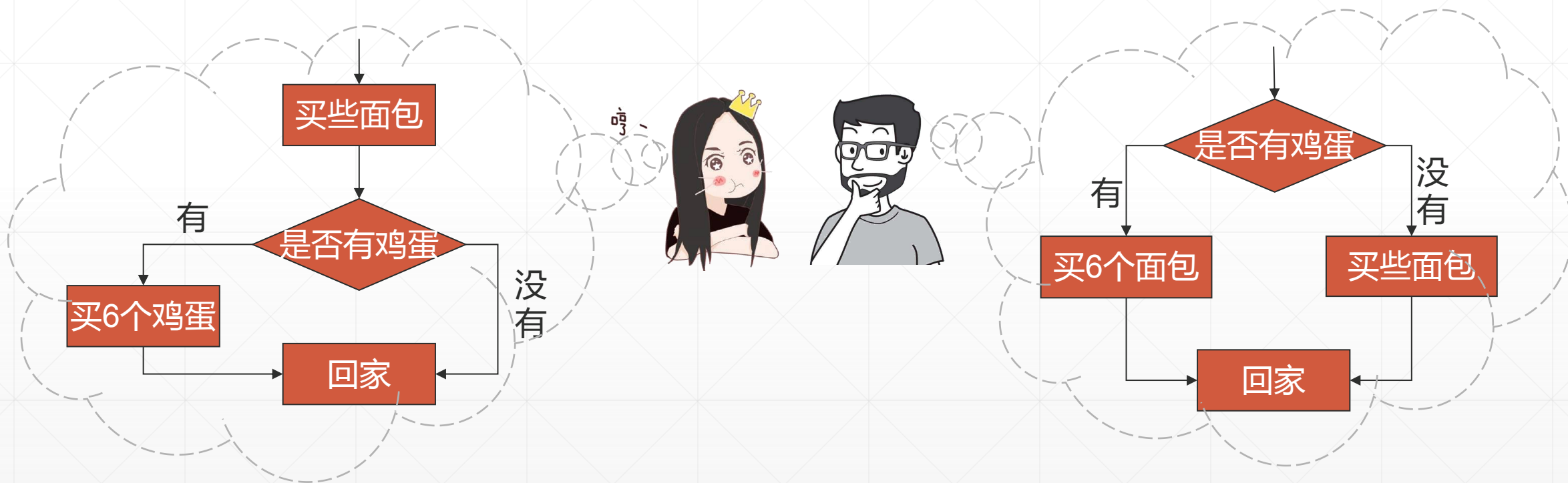
程序员的笑话

程序员的老婆吩咐他去商店买东西：你去附近的商店买些面包，如果有鸡蛋的话，买6个回来。

随后，老公买了6个面包回来。

他的妻子大吃一惊：你为什么买了6个面包？！

程序员回答：因为他们有鸡蛋。



流程图(flow chart)

利用各种方块图形、线条及箭头等符号来表达解决问题的步骤及进行的顺序；是算法的一种表示方式。

“标准作业流程”（SOP, Standard operating procedure）

- 企业界常用的一种作业方法，其目的在使每一项作业流程均能清楚呈现，任何人只要看到流程图，便能一目了然，有助于相关作业人员对整体工作流程的掌握。

程序流程图(program flow chart)

- 表示程序中的处理过程。

制作流程图的优点



所有流程一目了然，工作人员能掌握全局。




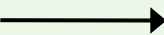
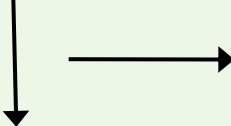

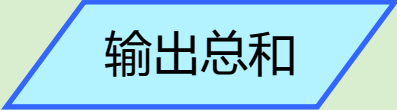

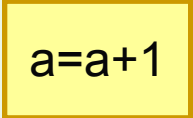


更换人手时，按图索骥，容易上手。


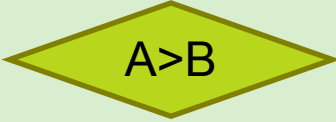

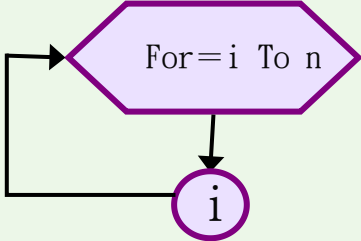

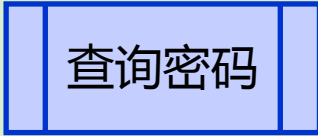




所有流程在绘制时，很容易发现疏失之处，可适时予以调整更正，使各项作业更为严谨。

流程图的基本符号

	名 称	意 义	范 例
	开始 (Start) 终止 (End)	表示程序的开始或结束	 
	路径(Path)	表示流程进行的方向	
	输入(Input) 输出(Output)	表示数据的输入或结果的输出	
	处理(Process)	表示执行或处理某一项工作	

流程图的基本符号

	名 称	意 义	范 例
	决策判断 (Decision)	针对某一条件进行判断	
	循环 (Loop)	表示循环控制变量的初始值及终值	
	子程序 (Subroutine)	用以表示一群已经定义流程的组合	
	文件(Document)	指输入输出的文件	

流程图的基本结构

顺序结构 (Sequence)

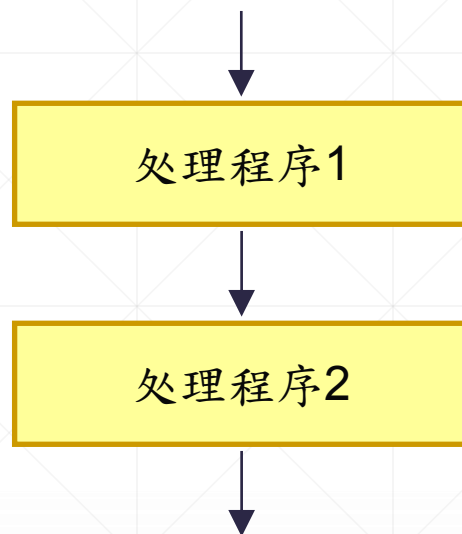
选择结构 (Selection)

- 二元选择结构 (基本结构)
- 多重选择结构

循环结构 (Iteration)

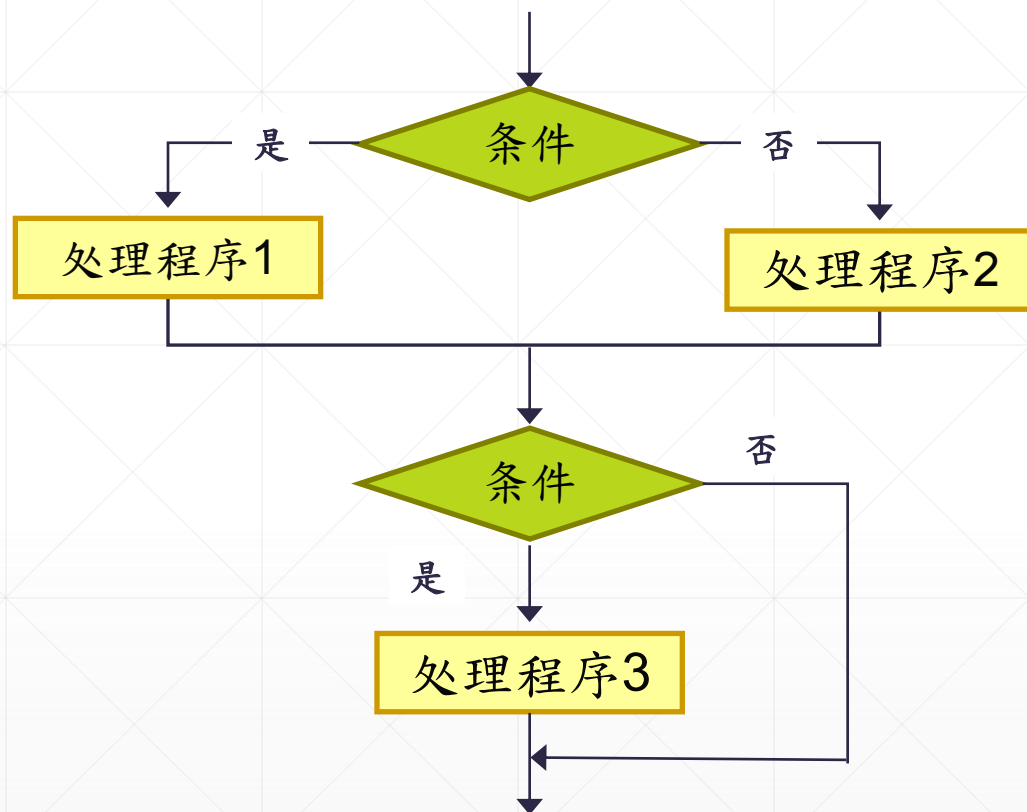
- while-do结构
 - do-while结构
-

顺序结构 (Sequence)



意义：处理程序顺序进行。

二元选择结构（基本结构）

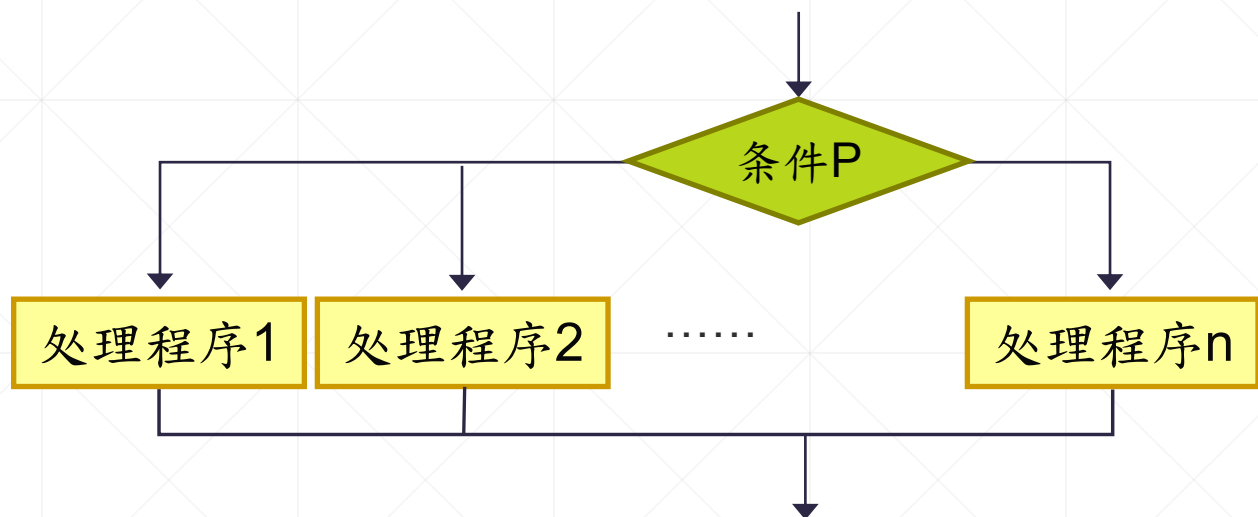


语法：

```
if (条件) {  
    处理程序1;  
}  
else {  
    处理程序2;  
}  
if (条件) {  
    处理程序3;  
}
```

意义：流程依据某些条件，依条件是否成立，分别进行不同处理程序。

多重选择结构

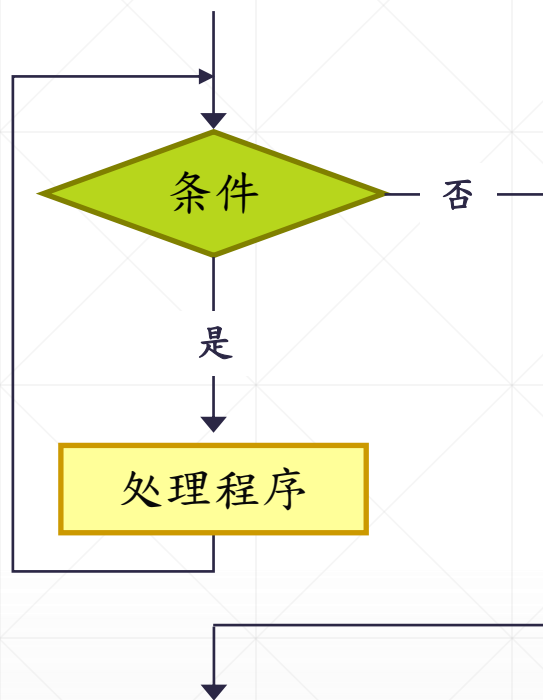


语法：

```
switch (条件) {  
  case p=1:  
    处理程序1;  
  case p=2:  
    处理程序2;  
  ...  
  case p=n:  
    处理程序n;  
}
```

意义：流程依据某些条件，在不同的条件成立时，分别进行不同处理程序。
例如条件P=1时，进行处理程序1。条件P=n时，进行处理程序n。

while-do结构

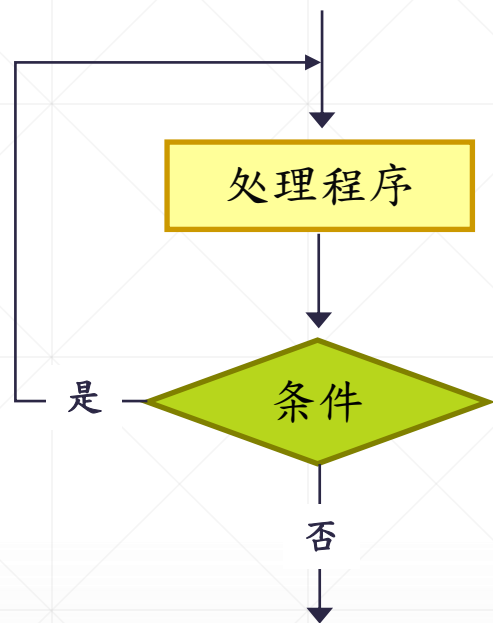


语法：

```
while (条件) {  
    处理程序;  
}
```

意义：依据条件是否成立，决定执行的情况。当条件成立时，不断重复执行处理程序，直到停止执行的条件成立后，即离开重复执行，至下一个流程。

do-while结构



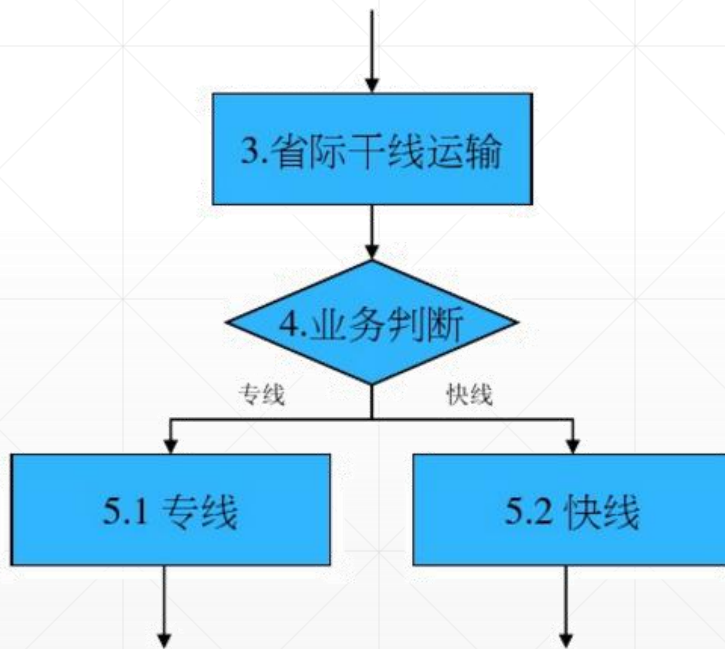
语法：

```
do {  
    处理程序;  
}  
while (条件);
```

意义：重复执行处理程序，直到条件变成假（false）为止。

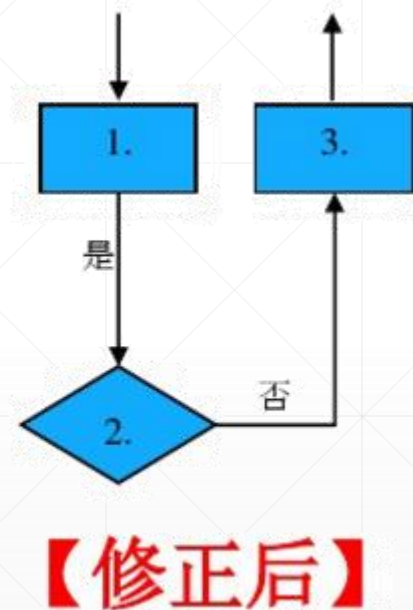
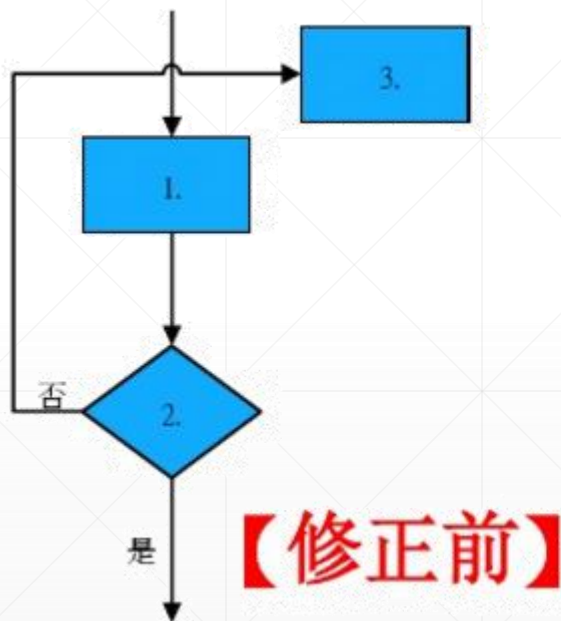
流程图绘制原则

1. 各项步骤有选择或决策结果，如“可/否”、“通过/不通过”或其他相对文字时，请检查校正流程是否有遗漏，以避免悬而未决的状况。
2. 流程图符号绘制排列顺序，为由上而下，由左而右。
3. 处理程序可用阿拉伯数字，从1开始，依处理程序排列顺序编号，并以文字依处理程序功能命名。



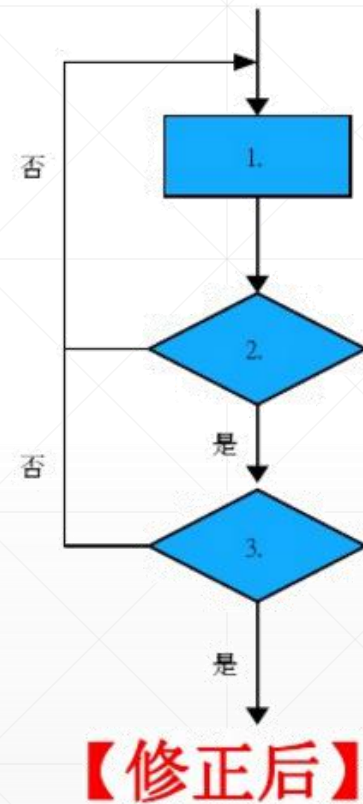
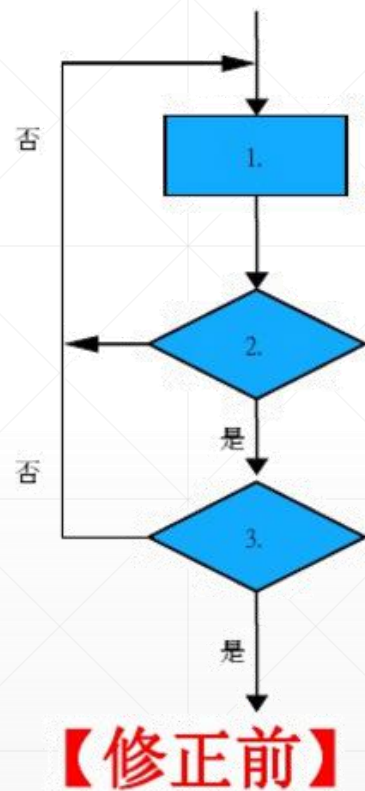
流程图绘制原则 (续)

4. 相同流程图符号宜大小一致。
5. 路径符号宜避免互相交叉。



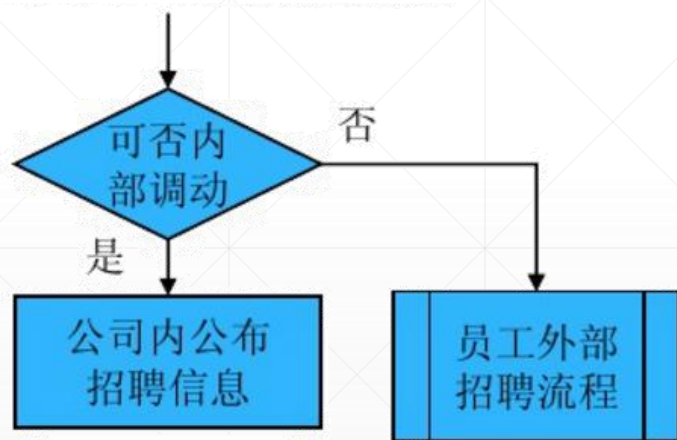
流程图绘制原则 (续)

6. 同一路径符号的指示箭头应只有一个。



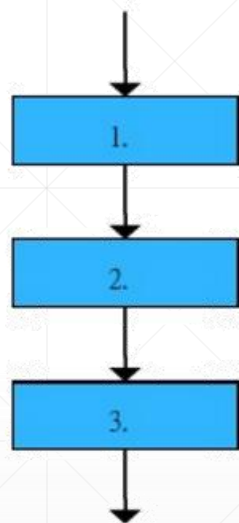
流程图绘制原则 (续)

7. 开始符号在流程图中只能出现一次，但结束符号则不限。若流程图能一目了然，则开始符号和结束符号可省略。
8. 选择结构与重复结构的选择或决策条件，文字叙述应简明清晰，路径加注“是”、“否”或其它相对性文字指示说明。
9. 流程图中若有参考到其他已定义流程，可使用已定义处理程序符号，不必重复绘制。

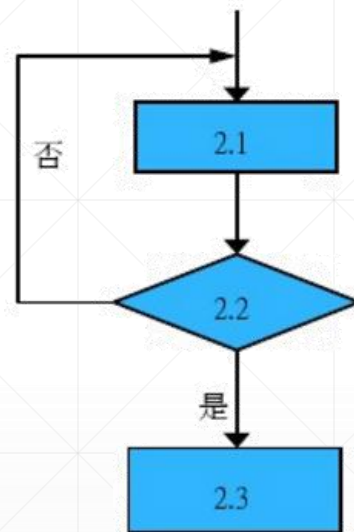


流程图绘制原则 (完)

10. 流程图若一页绘制不下，可以使用分级分页绘制方式，并在处理程序编号上表示其级别。

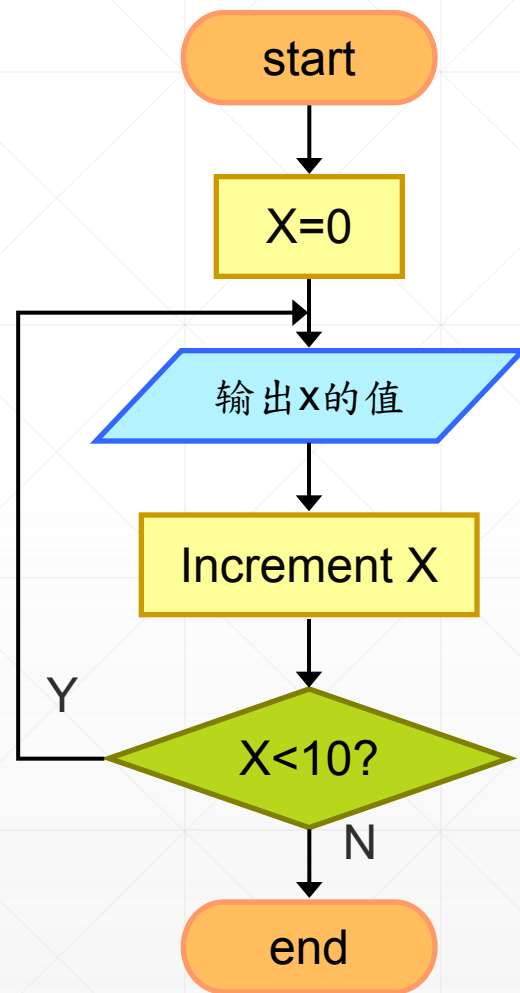


【第一页】



【第二页：处理程序2的分级分页绘制】

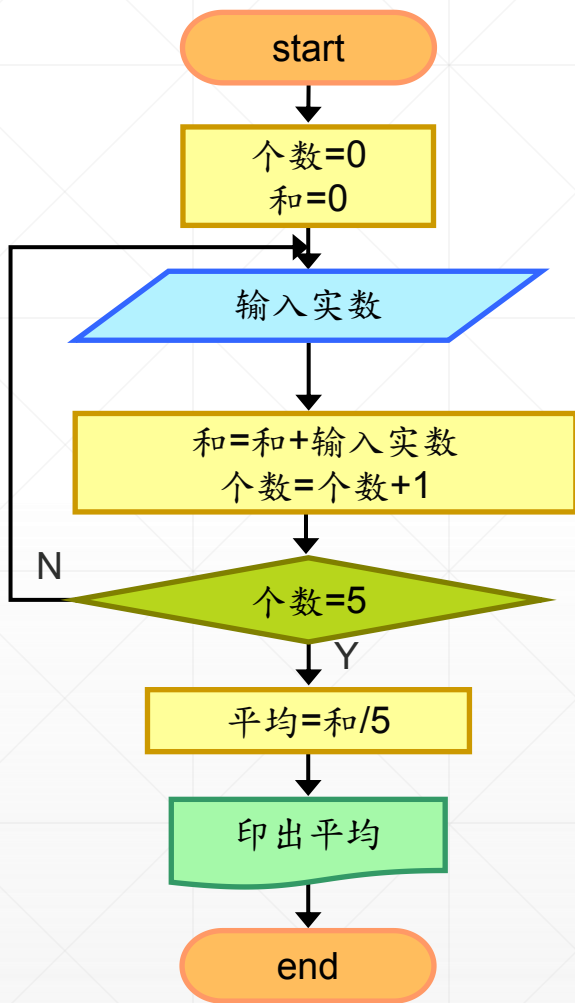
示例：输出0~9的数



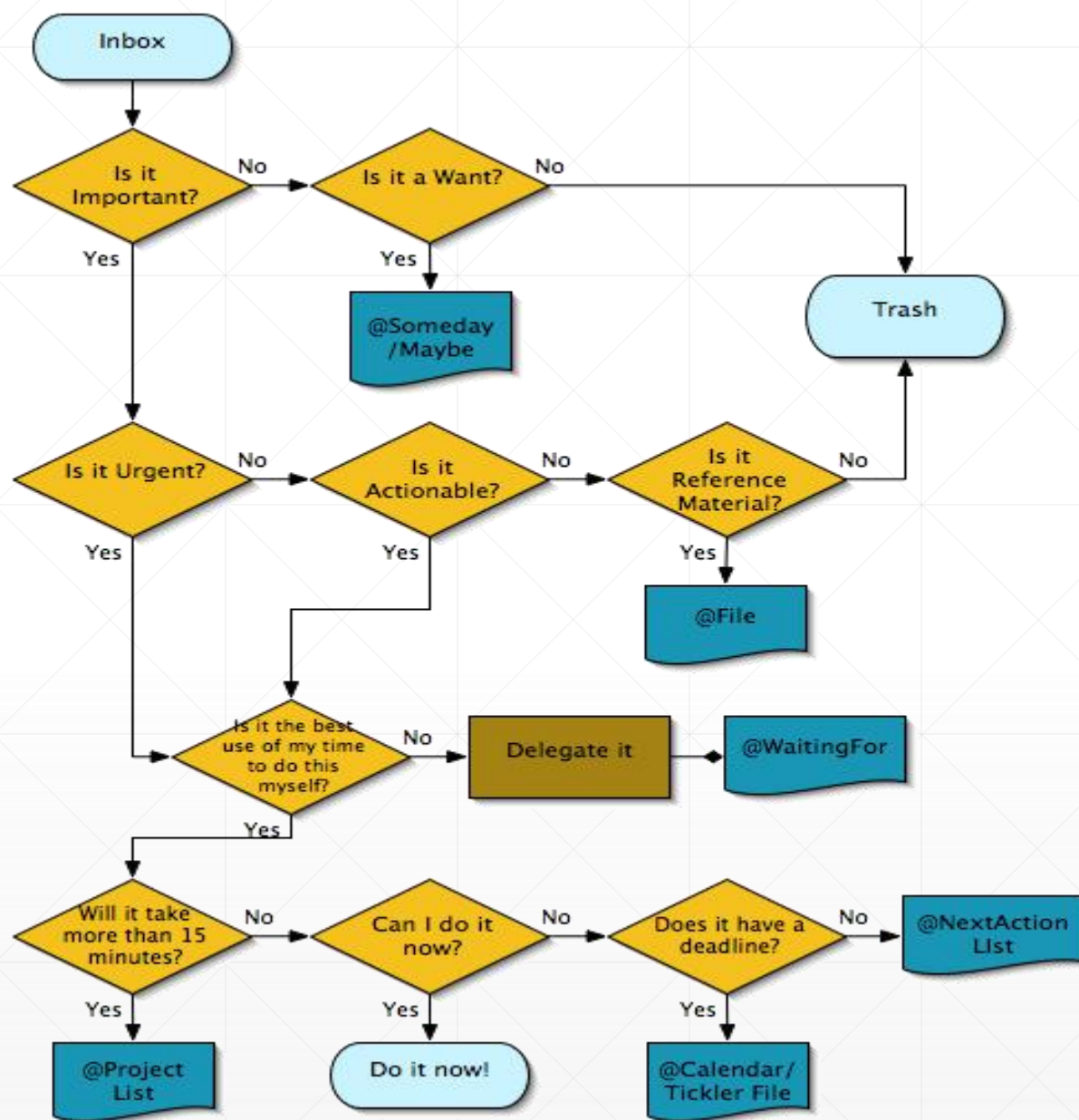
输出结果

0123456789

示例：输入5个实数，计算平均值，然后再打印该平均值



示例：收邮件





授课教师：蓝天 电子邮箱：lantian1029@uestc.edu.cn