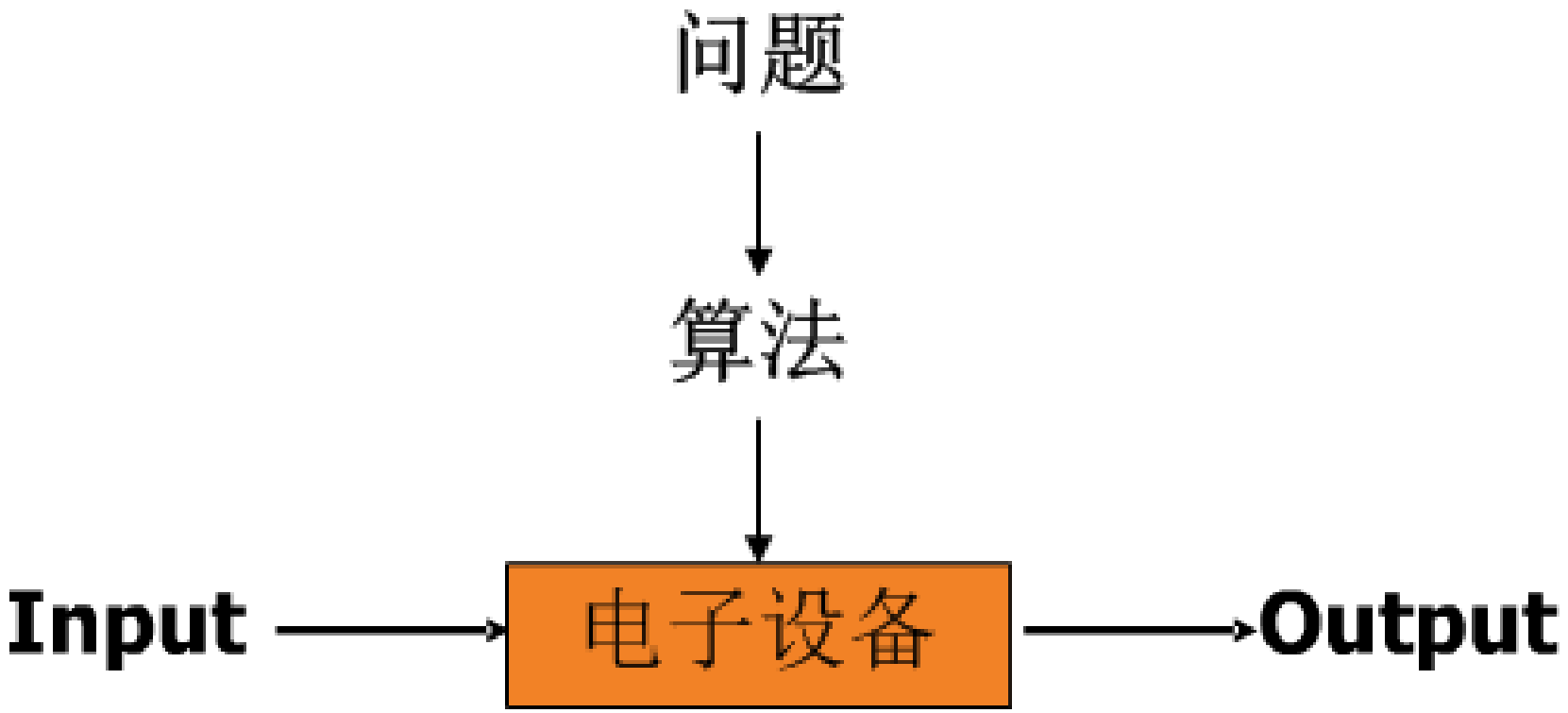
1.1 算法与程序

1 算法定义及其特性

算法：是将问题的输入转化为输出的一系列计算或操作步骤.

“算法是任何定义好的计算程式，它取某些值或值的集合作为输入，并产生某些值或值的集合作为输出。”



注意：虽然绝大多数算法最终会靠计算机来执行，但算法概念本身并不依赖于这一假说。

算法描述举例

例：求两个不全为0的非负整数m, n的最大公约数gcd(m,n)的欧几里德算法描述:

1.如果n=0,返回m的值作为结果，过程结束；
否则，进入第二步；

2.用n去除m，将余数赋给 r;

3.将n的值赋给m，将r的值赋给n,返回第一步。

计算机算法与人工算法

例如 求定积分: s= $\int_a^b f(x)dx$

人工处理步骤为
- **找出*f*(x)的源函数F(x)**
- **利用牛-莱公式:**s=F(b)-F(a)

计算机算法：计算定积分采用数值积分的方法,得到一个近似解.

- 有些问题没有计算机算法.
- 有些问题计算机算法与人工算法不同.

算法的定义因看待的角度不同而不同

- 哲学家：算法是解决一个问题的抽象行为序列。
- 码农：算法是一个计算过程，它接受一些输入，并产生某些输出。
- 高大上：算法是解决一个精确定义的计算问题的工具。

核心：算法是解决问题的办法和法则，算法必须能够让人一步一步照着执行。

算法的特征

1. 有穷性

一个算法须在执行有限个运算步后终止，每一步必须在有限时间内完成. 实际应用中，算法的有穷性应该包括执行时间的合理性.

程序是算法的程序设计语言的具体实现.可不满足性质1.

一个算法面向一个问题，而不是仅仅求解一个问题的实例

操作系统程序：是一个在无限循环中执行的程序，而不是一个算法。

2. 确定性

算法的每一步骤必须有确定的含义，对每一种可能出现的情况，算法都应给出确定的操作，不能有多义性.

例如　计算分段函数 $f(x)=\begin{cases}1 & x>100\\ 0 & x<10\end{cases}$

算法描述：输入变量*x*，

- ✓**若*x*大于100的数，输出1；**
- ✓**若*x*小于10的数，输出0.**
- ✓**输入10<=*x*<=100,则算法在异常情况下,执行结果是不确定的.**

3. 能行性

算法中的每个步骤是能实现的，如 x/0；负数开方...

算法的执行结果达到预期目的，正确，有效.

4. 输入

有0个或多个输入项.

5. 输出

算法产生至少有一个输出项

2.算法设计过程(程序设计过程)

1. 问题的陈述

理解问题，并用科学规范的语言把所求解问题进行准确的描述,包括所有已知条件和输出要求.

2. 建立数学模型

通过对问题分析,找出其中所有操作对象以及对象之间的关系,并用数学语言加以描述. 对非数值型解法来说,数学模型通常是链表,树,图,集合等数据结构.

3. 算法设计

根据数据模型，给出求解问题的一系列步骤，且这些步骤可通过计算机的各种操作来实现.

1. 不含语法错误；
2. 对几组输入数据能够得到满足规格需求的解；
3. 对精心选择的典型数据能够得到满足规格需求的解；
4. 对一切合法的输入数据都能得到满足规格需求的解。

5. 算法的程序实现

将一个算法描述正确地编写成机器语言程序.

6. 算法分析

对执行该算法所消耗的计算机资源进行估算，对数值型算法还需分析算法的稳定性和误差等问题.

计算机资源中最重要的是时间和空间资源，执行一个算法程序需要的时间和占用的内存空间分别称为**算法的时间复杂度和空间复杂性** .

- **算法的复杂性分析具有极重要的实际意义。**
  - **有许多实际应用问题,理论上是有计算机解的,但由于求解所需的时间或或空间耗费巨大,如成千上万年,以至于实际上无法办到;**
  - **对有些时效性很强的问题,如实时控制,即使算法执行时间很短,只有一两秒,也可能是无法忍受的。**

- 算法的评价

--modularity模块化

--user-friendliness用户的友好性

--correctness正确性

--programmer time 程序员的时间

--maintainability 可维护性

--simplicity 简单性

--functionality 功能性

--extensibility 扩展性

算法评估标准：正确性, 运算时间, 占用空间, 简单性, 健壮性

3.算法的描述

描述算法的方式一般有三种:自然语言,流程图,伪代码语言。 伪代码描述介于自然语言与程序设计语言之间。

例：从键盘上输入一个正数，然后计算它与10的和。

1. 自然语言描述算法：

①. 输入a;

②. 判断a是否大与0;

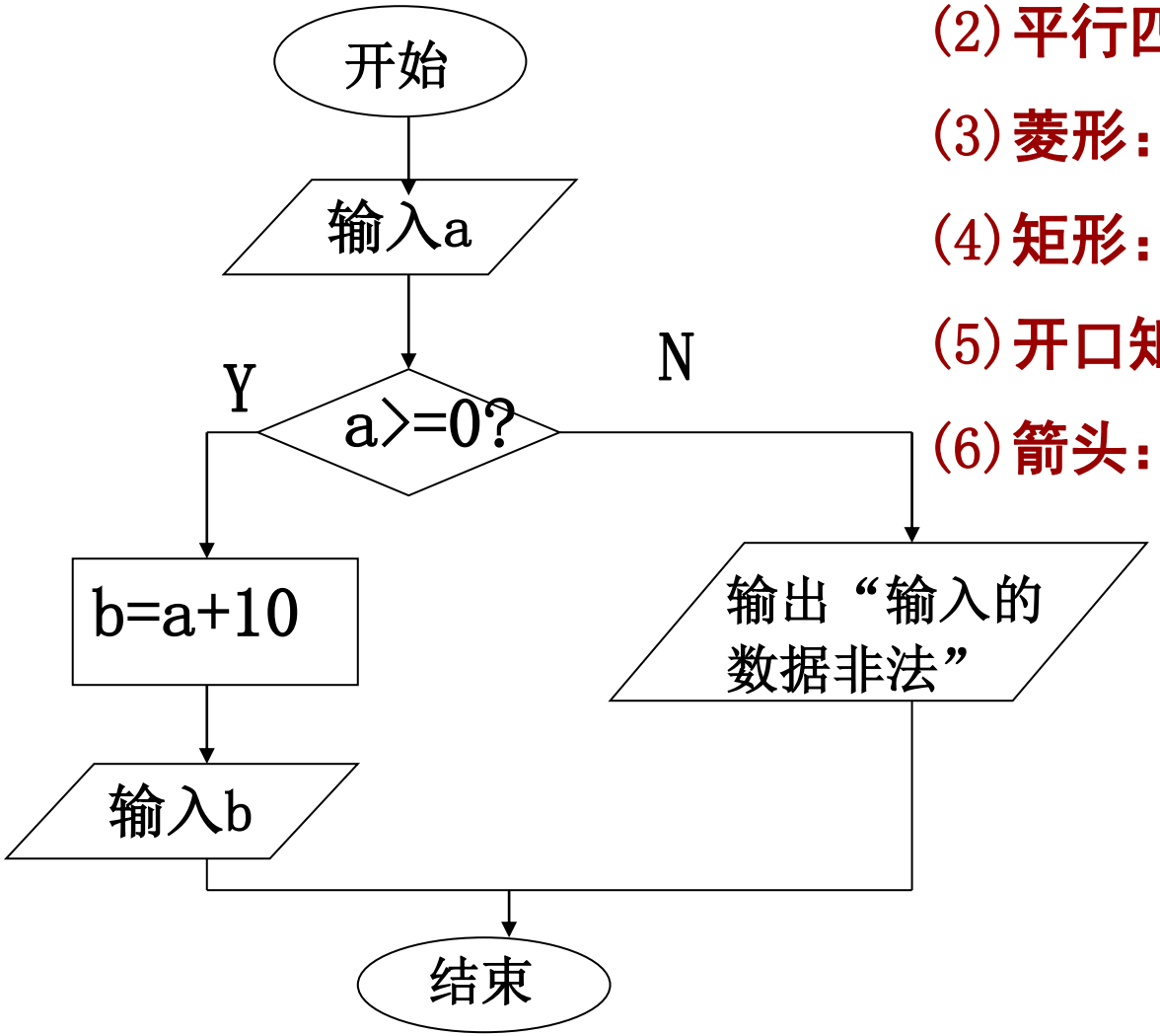
③. 如果a大于0，则执行④，否则去执行⑥;

④. 计算b，b=a+10 ;

⑤. 在屏幕上输出b的值，并结束算法。

⑥. 在屏幕上输出“输入的数据不合法。”语句，并结束算法。

用流程图描述算法：



(1) 椭圆：表示起始、终止

(2) 平行四边形：输入输出

(3) 菱形：判断

(4) 矩形：执行表达式和赋值

(5) 开口矩形：注释框

(6) 箭头：数据流向

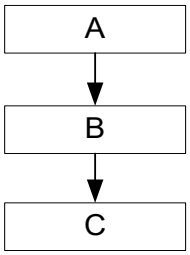
3．用伪代码描述算法：

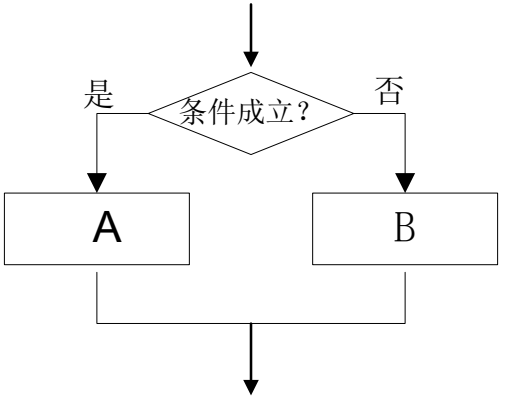
```
{
  输入a的值;
  If (a>0) {b=a+10; 输出b的值; }
  Else 输出“输入的数据不合法。”;
  }
```

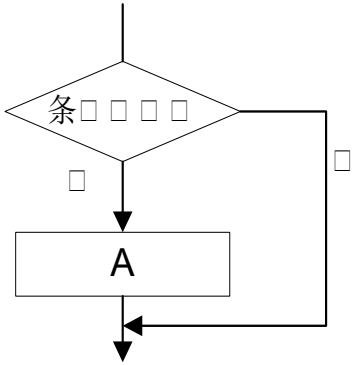
缩进表示块结构,while循环退出后计数器保持其值,//表示注释, i=j=e表示将e的值赋给i和j,一般不使用全局变量,过程调用按传值参数处理

4. 算法结构

任何算法都可由顺序结构、选择结构、循环结构这三块“积木”通过组合和嵌套表达出来，遵循这种方法的程序设计，即为结构化程序设计。







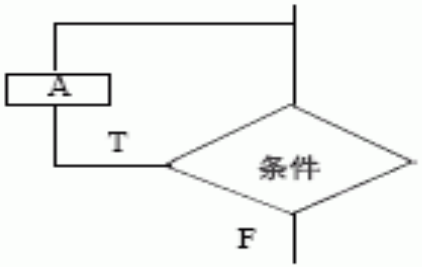


图1-6 while 型循环流程图

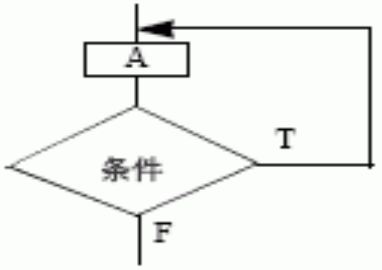


图1-7 do-while型循环流程图

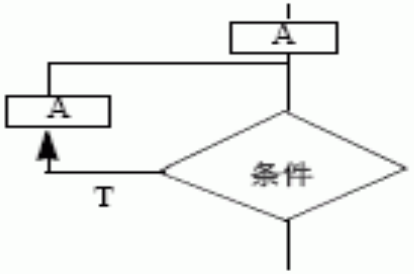


图1-8 do-while 型循环转换为while型循环

5. 算法分类

从解法上

- 数值型算法：算法中的基本运算为算术运算.
- 非数值型算法：算法中的基本运算为逻辑运算.

从处理方式上

- 串行算法：串行计算机上执行的算法.
- 并行算法：并行计算机上执行的算法.

本课程主要介绍非数值型的串行算法.