

算法与流程图

广州协和学校信息技术科

2024年11月

通过编程解决问题的步骤

- 首先通过分析找出解决问题的方法和步骤;
- 然后将待处理的信息用计算机能识别的方式表示, 使之成为计算机能处理的数据;
- 最后编写并调试程序, 使之无误并能顺利运行。

算法概述

- 算法 (Algorithm) 是指解题方案的准确而完整的描述, 是一系列解决问题的清晰指令, 算法代表着用系统的方法描述解决问题的策略机制。
- 从计算机角度而言, 程序可视为数据结构和算法的集成, 体现在以下两方面:
 - (1) 对数据的描述
 - (2) 对数据的具体操作。

算法的特征

- 算法是解决问题的有限步骤序列，其设计和运行必须满足以下五个基本特征：

有穷性：算法必须在有限的步骤内完成。如果算法存在无限循环或者无法停止，就不能称之为算法。例如：计算从1到100的和，一个正确的算法会在执行100次加法后终止，而不会永远运行下去。

确定性：算法的每一步操作都必须明确、清晰，没有任何歧义。在相同的输入条件下，每次运行算法时，结果应该是完全相同的。例如：一个排序算法在给定相同的输入数组时，必须始终输出相同的排序结果。

有零个或多个输入：输入是算法的初始数据，可能来自用户、文件或其他系统。

有一个或多个输出：一个算法必须至少有一个输出，即算法的处理结果。输出是算法的目的所在，没有输出的过程不构成算法。

有效性：算法中的每一步操作都应该是可以有效执行的。也就是说，算法所涉及的每个步骤都可以通过基本计算在有限时间内完成。

算法的表示

- 自然语言

用人类常用的语言（如中文、英文）按照一定的逻辑描述算法的执行步骤。

- 程序流程图

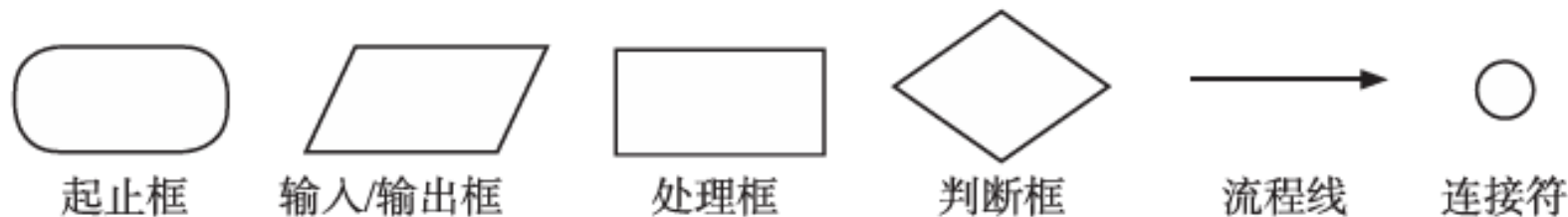
通过图形化的方式，用特定的符号（如矩形、菱形、箭头等）来表示算法的逻辑流程。

- 程序代码

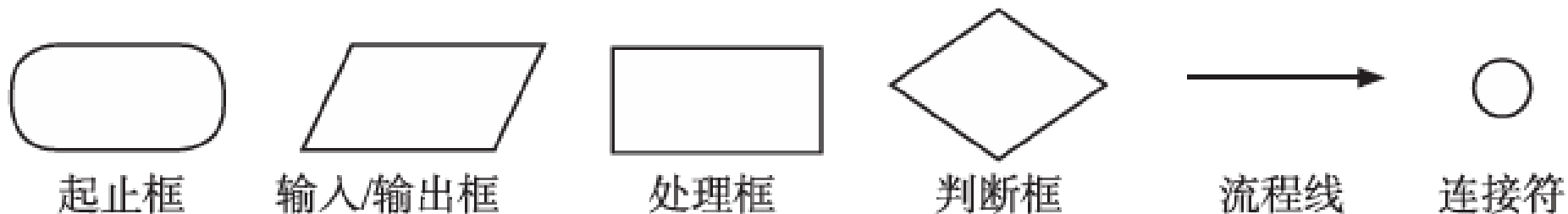
用编程语言（如Python、C++等）编写的精确指令来实现算法。

使用流程图描述算法

- 流程图也称为程序框图，它是算法的一种图形化表示方法。与使用自然语言描述算法相比，用流程图描述算法形象、直观、更容易理解。
- 对于一些复杂的算法，直接编写代码很困难，我们可以先用流程图描述算法，然后根据流程图写出程序代码。
- 为了使算法的流程图看起来更加直观，需要对流程图的表示方法进行规范，目前最常用的流程图规范是由美国国家标准学会制定的一系列流程图符号。



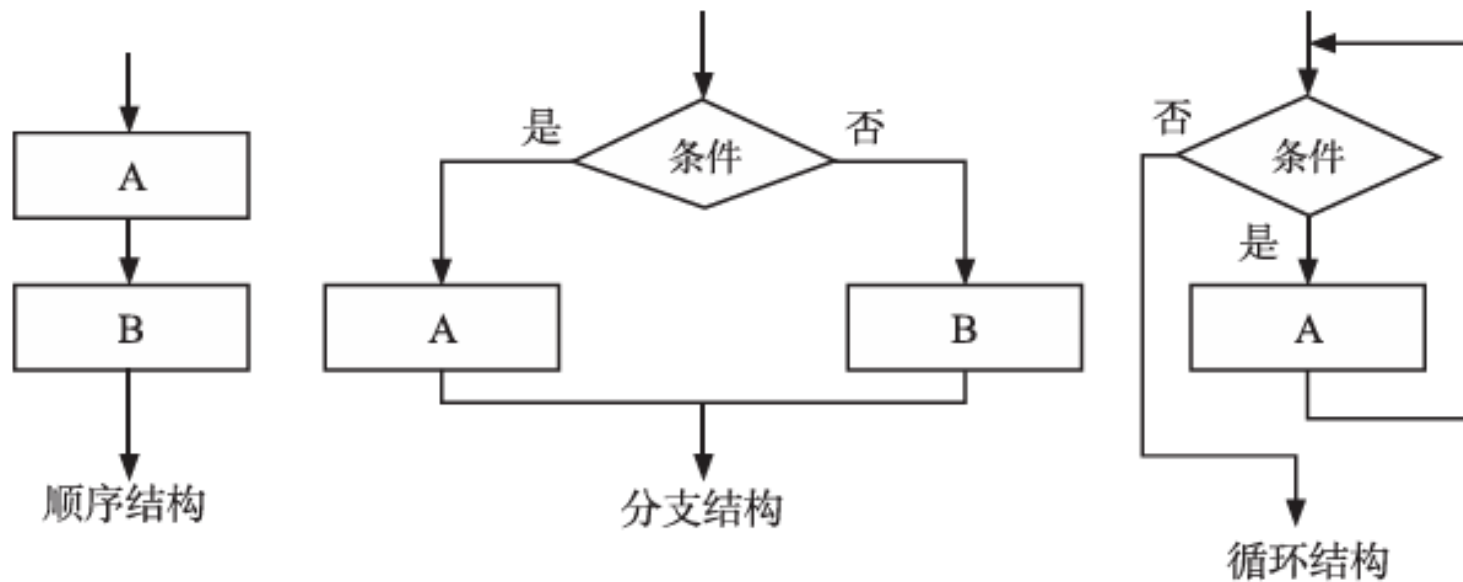
使用流程图描述算法



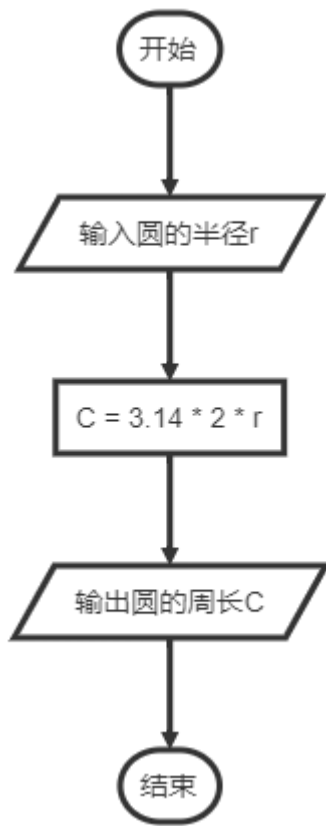
- ◆ 起止框：表示一个算法的开始和结束。
- ◆ 输入/输出框：表示从外部输入数据到计算机内部或者从计算机内部输出数据到计算机外部。
- ◆ 处理框：表示操作的内容。
- ◆ 判断框：表示判断的条件。满足条件，执行标识为“是”的路径；不满足条件，则执行标识为“否”的路径。
- ◆ 流程线：指向算法运行的方向。
- ◆ 连接符：表示流程图的接续。在相互联系的流程图内，流程线将在具有相同数字或字母的另一连接符处继续下去。

三种基本结构的流程图

算法的三种基本结构是顺序结构、分支结构和循环结构。

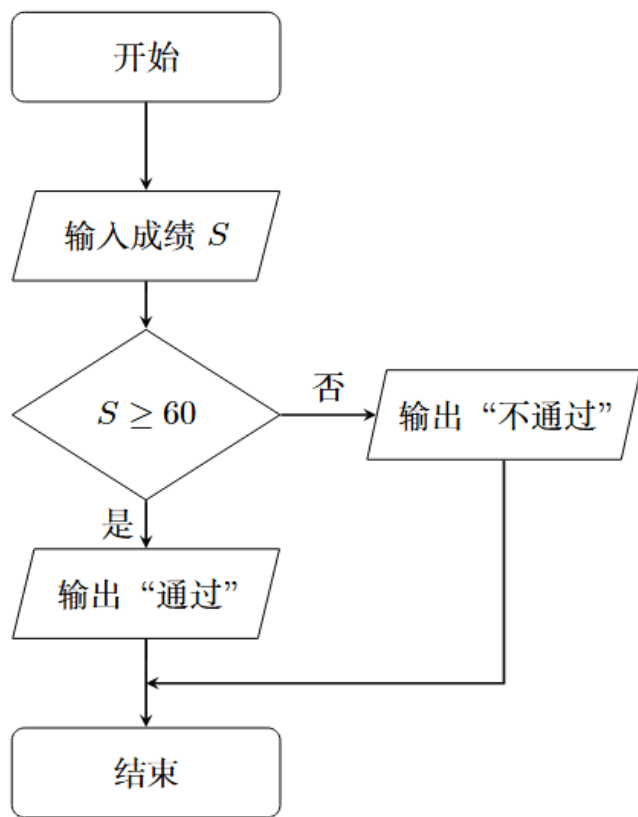


例题1：计算圆的周长（顺序结构）



```
r = int(input())  
C = 3.14 * 2 * r  
print(C)
```

例题2：成绩计算（分支结构）



```
S = int(input())
if S >= 60:
    print("通过")
else:
    print("不通过")
```