

经全国中小学教材审定委员会2004年初审通过  
普通高中课程标准实验教科书

# 信息技术 · 选修1

## 算法与程序设计

SUANFA YU CHENGXU SHEJI

祝智庭 主编



中国地图出版社



经全国中小学教材审定委员会2004年初审通过  
普通高中课程标准实验教科书

**信息技术** · 选修1

# 算法与程序设计

SUANFA YU CHENGXU SHEJI

祝智庭 主编



中国地图出版社

本套教材主编：祝智庭  
本套教材副主编：刘观武 任友群  
本册教材主编：祝智庭  
本册教材副主编：高淑印

普通高中课程标准实验教科书

信息技术·选修1

**算法与程序设计**

祝智庭 主编

---

**中国地图出版社 出版**

北京市宣武区白纸坊西街3号 邮编:100054

网址: <http://www.sinomaps.com>

北京市北关闸印刷厂印刷 新华书店发行

---

开本: 890毫米×1240毫米 16开 印张: 10 字数: 213千字

2005年1月第1版 2006年6月第4次印刷

ISBN 7-5031-3814-9/G·1513

---

定价:16.06元(含1张CD-ROM)

**版权所有 侵权必究**

图书与光盘如出现质量问题,请及时与本社联系。

## 编写说明

本套教材根据教育部《普通高中技术领域课程标准》(信息技术部分)编写,供高中阶段学习使用。本套教材共分6册:必修模块为《信息技术基础》,选修模块依次为《算法与程序设计》、《多媒体技术应用》、《网络技术应用》、《数据管理技术》和《人工智能初步》。

整套教材以“知识引领、活动穿插;任务引领、知识渗透;工具支持、资源配套;评估跟进、形式多样”为编写思路。从解决学生日常生活、学习中的实际问题入手,运用信息获取、加工、管理、表达与交流的基本方法,在以主题活动、探究性学习等多种形式的学习过程中逐步提升学生的信息素养,从而实现知识与技能、过程与方法、情感态度与价值观三个方面的培养目标。为了支持学习和创作表达的过程,有利于发展性评价,教材中引入了基于网络环境的“电子学习档案袋”。每册教材配学习光盘(CD-ROM),以便于使用光盘中的资源,且在教材中作了相应的提示。通过信息技术学习网站(<http://itedu.tjyy.com.cn>)提供互相交流的平台,并及时更新和拓展教学资源。学习的测评由电子学习档案袋、电子作品和在线考试三种方式组成,网站提供了上传电子作品的应用软件、电子学习档案袋安装软件和支持在线考试的相关资源。

本册教材为选修模块1,供36学时使用。

本套教材由华东师范大学教授、博士生导师祝智庭任主编,特级教师刘观武和华东师范大学副教授任友群博士任副主编。

本册教材主编祝智庭,副主编高淑印,编者方文祺、齐国英、黄福铭、刘观武。

欢迎广大师生通过电子邮件([infotech@sinomaps.com](mailto:infotech@sinomaps.com)或[tjyy@tjyy.net](mailto:tjyy@tjyy.net))与我们交流,提出意见和建议,指出差错或不足,共同推动信息技术课程教材建设。

天津市教育教研室

中国地图出版社

2004年6月

# 前言

《数字化生存》的作者尼葛洛庞帝 (Negroponte) 说：“计算，不再只与计算机有关，它决定着我们的生存”。

计算与算法之间、程序与计算机之间，是唇齿相依的关系。计算机离不开程序的控制，程序的核心是算法的设计。

同学们，你们听说过吗？1997年，IBM公司的“深蓝”(Deep Blue) 计算机战胜了国际象棋世界冠军卡斯帕罗夫，引起了人们“今天输了最伟大的棋手，明天将输掉什么”的惊呼。而“深蓝”战胜人的基础就是算法与程序设计。

实际上算法的起源比电子计算机的出现要早很多。当同学们欣赏动画片、游戏等计算机实现的三维动画效果时，你们可知道这里面浸透着科学家上百年的算法研究成果？

当今指纹、虹膜、DNA等生物特征识别技术的广泛应用，金融电子化与消费的增长，网络加密与反加密的斗争，对月球、火星的探测……你知道这些新生事物的核心包括算法设计吗？

算法与程序设计造就了今天计算机的“智慧”和“能耐”，而创造这些神奇算法的人们，必将在算法的天地里，用程序代码编织出梦幻般的未来。

**现在，让我们来共同探索这个神奇的算法与程序设计的世界，一起来感受人类智慧的辉煌吧！**



# 目 录

## 第一单元 走进编程

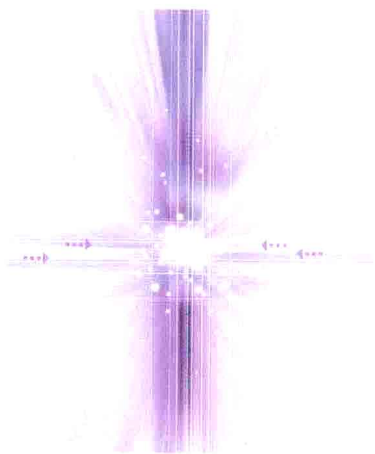
1

- 第一节 解决问题的一般方法—— 2
- 第二节 解决问题的算法设计—— 6
- 第三节 算法的程序实现—— 14
- 第四节 程序设计语言简介—— 22

## 第二单元 程序设计基础

25

- 第一节 数据及其运算—— 26
- 第二节 顺序结构—— 34
- 第三节 分支结构—— 42
- 第四节 循环结构—— 50
- 第五节 方法与模块化程序设计—— 64
- 第六节 面向对象程序设计初步—— 75
- 第七节 图形用户界面的程序设计—— 91



## 第三单元 算法与问题解决

107

- 第一节 解析法与问题解决—— 108
- 第二节 穷举法与问题解决—— 120
- 第三节 递归法与问题解决—— 127
- 第四节 排序与查找—— 134



## 第四单元 尝试软件开发

143

- 第一节 项目策划—— 144
- 第二节 项目实施—— 149

## 中英文对照表

153

# 第

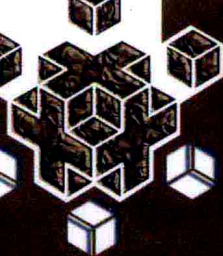
# 1

## 单元 · 走进编程

“授人以鱼不如授人以渔”。掌握一门编程语言不再令人感到神秘而高不可攀。但是，要想掌握一种解题方法，却不是每个人都能轻易做到的。人们编写程序就是为了解决问题，只有通过预先分析问题来确定必须达到的性能目标，才有希望找到正确的问题解决方案。

在刚踏入编程这个门槛时，我们先来了解计算机能解决哪些问题，并在经历计算机解决问题的基本过程中，掌握用算法去解决实际问题的基本方法，最终学会把复杂的问题转化为计算机能处理的程序。





## 第一节 解决问题的一般方法

日常生活中我们会遇到各式各样的问题，随着计算机技术的发展，大多数问题都可以使用计算机来解决。其中，有些问题可以使用现成的计算机工具软件解决，有些问题则需要编写特定的计算机程序才能解决。本节我们将学习解决问题的一般方法，特别要学会辨别哪些问题需要编写计算机程序来解决，并经历计算机解决问题的基本过程。

我们在解决问题时一般采取的步骤是：通过向老师或父母请教，上网或去图书馆查资料，初步形成自己的观点；设计解决方案并验证或反复验证观点；最终得到问题结论。即要经历猜测、验证和结论等过程。

21世纪，信息技术为我的世界带来了巨大的变革，计算机成为解决问题的重要工具。利用计算机解决问题的一般过程通常是先分析问题，确定解决问题的方案，根据方案来设计解决问题的具体步骤，然后交给计算机执行，最终解决问题。

让我们一起来经历利用计算机解决问题的过程吧！

### 一 分析问题

现实生活中，我们经常会遇到一些需要解决的问题。例如，选择最近的上学路线，如何解数学计算题，填报升学志愿书时选择最满意的学校，等等。请大家找出一些问题，在小组内讨论一下，把适合用计算机解决的问题提出来，并填写在下面的横线上：

问题 1: \_\_\_\_\_

问题 2: \_\_\_\_\_

问题 3: \_\_\_\_\_

问题 4: \_\_\_\_\_

.....

例如，校医要对学生的体检表进行统计，试图分析出全校每个年级男、女同学的身高、体重及视力等身体状况。我们仅就其中一项指标来提出问题，如：某班同学谁最高？

解决问题首先要明确问题的目标和条件。本例中目标即求出“全班同学谁的身高最高”，条件是已知全班每一个同学的身高。

其次要理解问题，即要搞清楚问题涉及了哪些方面的知识。例如，本例是对体检表中的身高数值进行比较，比较数的大小是小学时就学过的数学问题。如果比较几个数的大小，直接观察可知结果。但是，在比较几十个甚至几百个数的大小时，就很难直接观察得



出结果，需要考虑求助工具（如计算机）帮助我们解决。这时，就需要了解计算机处理数据（如比较数的大小）方面的相关知识。

### 二 确定方案

解决方案是指采用什么工具、采取什么方法来解决问题。人们通常将问题的解决方案分为两大类：一类是可以执行若干个步骤就能得出问题结论的，这类问题的解决方案叫算法方案；另一类是不能通过若干个步骤直截了当地得出结论，而是需要反复验证才能解决的，这类问题的解决方案通常叫做启发式方案。在解决复杂的问题时，人们通常使用启发式方案。

计算机作为解决问题的重要工具，可以解决很多类型的问题。我们把能使用计算机解决的问题分为三类：

第一类：使用现有的工具软件可以解决的问题。

大多数需要用计算机解决的问题都属于这一类，例如上网查询“百慕大三角”的资料，编辑、打印班报，制作班级主页，收发邮件等。

解决这类问题的办法就是使用相应的计算机应用软件，如网络浏览器、文档编辑软件、主页制作软件、邮件收发软件等。这类问题的解决在《信息技术基础》一书中就已经学过了。

第二类：现成的工具软件解决不了，需要编写程序并通过执行若干个步骤才能解决的问题。

例如，计算 10000 以内的奇数和；查询某年的生肖；画 100 个同心圆；统计保龄球选手每人在每一局的积分等。

这类问题一般是：

- ◆ 计算型，指需要进行数学计算的问题；
- ◆ 逻辑型，指需要进行逻辑处理的问题；
- ◆ 需要反复执行一组计算型或逻辑型指令的问题。

这类问题主要使用算法方案解决，需要编写出解决问题的若干个具体步骤。对于人们来说处理这些问题既繁琐又耗时，但是用计算机处理起来就非常简单了，难点在于怎么设计具体的解决步骤，并用程序实现。

第三类：现成的工具软件不能直接解决问题，也需要编写程序，但不能通过既定步骤直接解决的问题。

例如：通过在线执行“PC 产品顾问 (<http://www.expertise2go.com>)”，为准备添置的个人电脑规划合理的软硬件配置方案；通过远程医疗网站进行诊病；人机对弈等问题。

这类问题不能由计算机通过既定步骤直接做出回答，具体问题需要具体分析，需要用启发式方案来解决。处理这类问题要利用计算机技术领域的人工智能技术。人工智能可以让计算机建立自己的知识库并学会人类的思维方式，从而使计算机解决问题的能力在某些方面将会和人相差无几。

关于“某班同学谁最高？”问题的解决方案可能有多种，那么我们试着列出一种：

如：让计算机来比较全班同学的身高值，找出最大值，即可知全班同学谁最高。

此方案经过若干个步骤后能直接得出结论，并且不需要反复验证，由此可以判定该方案属于算法方案，可以利用计算机编写特定的程序来解决。

请大家把前面提出的问题进行分类，找出相对集中的一些问题，然后从简单问题入手，共同探讨解决方法，并把解决问题的方案列出来：

问题 1 的解决方案：\_\_\_\_\_

问题 2 的解决方案：\_\_\_\_\_

问题 3 的解决方案：\_\_\_\_\_

这些解决方案能确定是属于哪一类解决方案吗？适合用计算机的哪一类工具来解决？能够通过有限的步骤得出问题的结论吗？还需要反复验证吗？

### 三 设计步骤

现在我们根据方案来设计具体的解决步骤，即进行算法设计。因为考虑到是用计算机解决问题，所以这些步骤必须能被计算机所执行。

本例解决方案的关键是对身高值进行比较，找出最大值，据此可以列出如下步骤：

步骤 1：输入第一个同学的身高值；

步骤 2：输入下一个同学的身高值；

步骤 3：比较两个同学的身高值；

步骤 4：选出较大值；

步骤 5：再输入下一个同学的身高值；

步骤 6：用这个同学的身高值和上述较大值进行比较；

步骤 7：再选出较大值；

重复步骤 5 至步骤 7，直到输入最后一个同学的身高值，比较后选出较大值；

步骤 8：该较大值即为全班同学的最高身高值。

### 四 程序设计

算法方案通常都要使用计算机编写特定的程序来解决问题，即用计算机能够理解的语言（程序设计语言）将算法表达成程序，得出最终结果，这就是程序设计。

问题分析是算法设计的基础，算法设计又是程序设计的基础。至于选择一种什么样的程序设计语言来解决问题并不是最重要的，关键是把握算法与程序设计的思想。本书后续内容将介绍如何使用算法和设计程序来解决问题。

例如，“某班同学谁最高？”的问题，按上述算法编写的程序运行结果如图 1-1-1 所示。





图 1-1-1 “某班同学谁最高?” 程序的运行结果



实践与思考

1. 请大家对课上提出的问题进行了筛选, 选出适合用计算机解决的问题, 列出其解决方案及具体的实施步骤。

2. 打开配套光盘“sc\unit1”, 执行小程序“查询生肖.exe”, 执行结果如图 1-1-2 所示。请回答下列问题:

- (1) 这个程序解决了什么问题?
- (2) 此问题如果不用计算机编程解决, 还有其他办法吗? 请写出解决步骤。

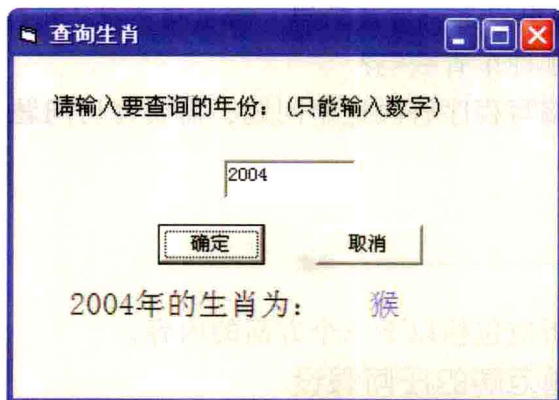
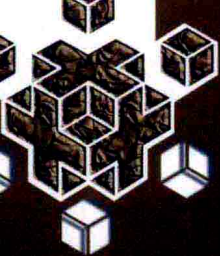


图1-1-2 查询生肖



## 第二节 解决问题的算法设计

本节我们将通过一个生活中的实例，学习如何设计算法并使用自然语言、伪代码和流程图等方法描述算法。

算法 (Algorithm) 就是解决问题的步骤序列。对问题的解决进行算法描述之前，通常要先分析问题，设计相应的算法，然后使用自然语言、伪代码或流程图等描述算法，将解决问题的思路表达出来。

### 一 提出问题

乘计程车已成为城市生活中一件很平常的事，计程车收费的主要依据是所行驶的里程数和计费标准，地区不同、车型不同，其计费标准也不相同。现有两种不同类型计程车，假设其计费标准分别为：

甲车 3 公里以下起步价是 10 元，3 公里以上（含 3 公里）每公里为 2 元；乙车 3 公里以下起步价是 8 元，3 公里以上（含 3 公里）每公里为 2.2 元。

你能比较一下乘坐哪种车省钱吗？

我们如果用计算机编写程序解决这个问题，需要经历问题分析、算法设计和程序实现三个过程。

### 二 分析问题

一般情况下问题分析应包括以下三个方面的内容。

#### 1. 能说明描述问题范畴的任何假设

在一个问题中，假设就是为了方便设计而假定的认为是正确的描述。例如，我们假设不考虑外在的因素，如等候时间造成的计费问题，只考虑相同里程数的费用。

#### 2. 已知信息

已知信息就是解决问题所需要的所有的已知条件，在描述问题时，要用“已知……”给出来。例如，已知两种计程车的计费标准和里程数。

#### 3. 目标求解

目标求解就是如何解决问题。还要具体说明问题解决后应该如何输出结果。例如，让计算机输出乘坐哪种车省钱。

综合这三方面内容，我们可以进行完整的问题描述。问题描述的格式不惟一，可以用文字描述、表格描述或图形描述等，只要把问题表述清晰即可。

我们对计程车的计费问题可以这样描述：比较两种计程车的计费情况，假设没有等候



时间；已知其计费标准和里程数；通过编程计算分别乘坐两种车的费用，并输出乘坐哪种车省钱。也可以使用表格形式描述问题，如表 1-2-1。

表 1-2-1 计程车计费问题的描述

假设	已知	求解
1. 甲、乙两种计程车	1. 甲车 3 公里以下起步价 10 元，以后每公里 2 元	1. 乘甲车的费用 (P1)
2. 无等候时间	2. 乙车 3 公里以下起步价 8 元，以后每公里 2.2 元	2. 乘乙车的费用 (P2)
3. 相同里程，费用低的省钱	3. 里程数为 $n$ 公里	3. 比较 P1 与 P2，如果： P1 < P2，输出 “ $n$ 公里时，乘甲车省钱” P1 > P2，输出 “ $n$ 公里时，乘乙车省钱” P1 = P2，输出 “ $n$ 公里时，两车费用相同”

你认为哪种描述更明确呢？如果你有更好的描述方法，请提出来供大家参考！

### 设计算法

算法设计是计算机解决问题的重要环节。算法一般具有可执行性、确定性、有穷性等特征，即算法的每一个步骤都是可以执行的，按照这个步骤一步步执行就能够得出正确的结论，并且能够终止。

#### 1. 算法的结构

##### (1) 输入数据

输入数据就是输入已知信息。在设计算法时，首先应具体说明计算机如何输入这些信息。

##### (2) 处理数据

处理数据就是列出计算机执行操作的指令序列。应该详细说明如何处理输入的数据，例如，根据里程数分别计算乘坐两种车的费用。

##### (3) 输出结果

算法的最后还要描述计算机如何输出信息。

#### 2. 算法的表示

表示算法的工具具有自然语言、伪代码、流程图、结构化流程图（又称框图，即 N-S 图）等等。这些工具不是程序设计语言，不能直接被计算机执行。使用它们的目的是把算法思想表达出来。

##### (1) 自然语言

用自然语言描述算法就是使用人们能读懂的简短语句对算法的步骤进行描述。例如，乘计程车一题的算法，用自然语言描述如下：

##### ① 输入数据：

输入乘甲车的费用 P1 为起步价 10 元

输入甲车 3 公里以后每公里的费用 X1 为 2 元

输入乘乙车的费用 P2 为起步价 8 元

输入乙车 3 公里以后每公里的费用 X2 为 2.2 元

输入里程数  $n$

## ②处理数据:

如果  $n$  小于 3: 则乘甲车的费用  $P1$  为 10 元, 乘乙车的费用  $P2$  为 8 元

如果  $n$  大于等于 3: 则乘甲车的费用为  $P1+X1 \times (n-2)$

乘乙车的费用为  $P2+X2 \times (n-2)$

## ③输出结果:

如果  $P1 < P2$ , 输出 “ $n$  公里时, 乘甲车省钱”

如果  $P1 > P2$ , 输出 “ $n$  公里时, 乘乙车省钱”

如果  $P1 = P2$ , 输出 “ $n$  公里时, 两车费用相同”

## (2) 伪代码

用伪代码描述算法就是采用一种类似于程序设计语言的代码来描述算法。伪代码没有统一的规定, 只要定义合理, 没有矛盾就可以。算法设计的步骤用伪代码表示的格式如下:

输入: . . . . .

输出: . . . . .

指令: . . . . .

. . . . .

. . . . .

. . . . .

. . . . .

本书中, 我们定义了 4 种伪代码的基本指令: 赋值指令、循环指令、条件指令和输出指令。

## ◆赋值指令

一般格式: 助记符  $\leftarrow$  表达式;

其中, 助记符是对数据命名的符号, 表达式是常量、助记符或是对助记符与常量进行运算的式子, 左箭头号 “ $\leftarrow$ ” 称为赋值号, 每条指令以分号结束。

语义: 对表达式进行计算, 并将这个值赋给助记符, 使助记符具有表达式的值。

例 1 把 10 元起步价赋给  $P1$ , 表示为:  $P1 \leftarrow 10$ ;

例 2 执行下面的指令, 计算  $n$ 、 $X1$ 、和  $P1$  的值。

$P1 \leftarrow 10$ ;

$X1 \leftarrow 2$ ;

$n \leftarrow 8$ ;

$P1 \leftarrow P1 + X1 * (n - 2)$ ;

分析: “ $P1 \leftarrow 10$ ;” 表示  $P1$  的值是 10; “ $X1 \leftarrow 2$ ;” 表示  $X1$  的值是 2; “ $n \leftarrow 8$ ;” 表示  $n$  的值是 8; “ $P1 \leftarrow P1 + X1 * (n - 2)$ ;” 表示先计算赋值号右边的表达式 “ $P1 + X1 * (n - 2)$ ” 的值, 即:  $10 + 2 * (8 - 2) = 22$ , 再把 22 这个值赋给  $P1$ , 所以  $P1$  的值是 22, 这里 “ $*$ ” 表示乘号。

结果: 执行完上述 4 行指令以后,  $n$ 、 $X1$  和  $P1$  的值分别为 8、2 和 22。

## ◆输出指令

一般格式: 输出 (表达式);

语义: 输出表达式的值, 表达式也包括表示文字信息的字符串, 字符串要用西文双引



号括起来。如果连续输出多项内容，各项之间用逗号分隔。

例如，输出“n 公里时，两车费用相同”，用输出指令表示为：

输出 (" n 公里时，两车费用相同")；

◆条件指令

一般格式：

```
if (条件表达式)
{
    指令序列 1
}
else
{
    指令序列 2
}
```

语义：若“条件表达式”成立，则执行“指令序列1”；否则（即“条件表达式”不成立），执行“指令序列2”。如果指令序列中只有一条指令，则指令序列头尾的花括号可以省略，简写为“if (条件表达式) 指令1;”。若“指令序列2”中无指令，那么“else”及其后边的花括号均省略。

条件表达式是用关系运算符来比较两个量的大小，例如  $x > 3$ 。条件表达式若成立，则其值为 true（真），若不成立则其值为 false（假）。两个或多个简单的条件表达式还可以组合成复杂的条件表达式，这时就要用逻辑运算符进行组合。关系运算符和逻辑运算符分别参见表 1-2-2 和表 1-2-3。

表 1-2-2

关系运算符

运算符	名称	表达式	语义
==	等于	$x == y$	x 与 y 相等时表达式值为 true, 否则为 false
!=	不等于	$x != y$	x 与 y 不相等时表达式值为 true, 否则为 false
>	大于	$x > y$	x 大于 y 时表达式值为 true, 否则为 false
<	小于	$x < y$	x 小于 y 时表达式值为 true, 否则为 false
>=	大于等于	$x >= y$	x 大于等于 y 时表达式值为 true, 否则为 false
<=	小于等于	$x <= y$	x 小于等于 y 时表达式值为 true, 否则为 false

表 1-2-3

逻辑运算符

运算符	名称	表达式	语义
&	与	$(x > y) \& (z != 0)$	x 大于 y 且 z 不等于 0 时表达式值为 true, 否则为 false
	或	$(x > y)   (z != 0)$	$(x > y)$ 和 $(z != 0)$ 中至少有一个为 true 时, 则表达式值为 true, 否则为 false

例如，如果  $P1 < P2$ ，输出“n 公里时，乘甲车省钱”，用条件指令表示为：

```
if (P1 < P2) 输出 (" n 公里时，乘甲车省钱")；
```

### ◆循环指令

在介绍循环指令之前，我们先来写一个求 4 个整数之和的算法。

```
a ← 输入第 1 个整数；  
b ← 输入第 2 个整数；  
c ← 输入第 3 个整数；  
d ← 输入第 4 个整数；  
sum ← a + b + c + d；  
输出 (sum)；
```

大家观察一下上面的算法，对于 a、b、c、d 的赋值，是不是都在重复同样的操作，如果要求 100 个整数的和，那么按照这种算法写下去，可以想像指令要有多长！有没有一个更好的办法代替这种重复操作呢？重复操作就是循环执行，因此可以用循环指令实现。

一般格式：

```
while (条件表达式)  
{  
    循环体  
}
```

语义：先判断“条件表达式”是否为真，若为假，则不执行“循环体”，结束循环指令；若为真，则执行“循环体”一次，然后再判断“条件表达式”，若为真，则再次执行“循环体”……直到“条件表达式”为假，结束循环指令。

设计循环指令时，循环体中必须包含能使“条件表达式”的值发生改变的指令，否则，大家想想会出现什么后果？例如，下面的算法指令是想求 4 个整数之和，请大家仔细阅读一遍算法，看看能取得预期结果吗？为什么？

```
输入：4 个整数  
输出：4 个整数的和  
指令：i ← 1；  
sum ← 0；  
while (i < 5)  
{  
    n ← 输入第 i 个数；  
    sum ← sum + n；  
}  
输出 (sum)；
```

在学习了上述 4 条伪代码指令后，我们用伪代码来描述计程车计费问题的算法如下：

```
输入：甲、乙两车的起步价及其每公里的车价和欲行驶的里程数  
输出：相同里程数时，乘坐哪种车省钱
```



```

指令：P1←甲车的起步价；
      P2←乙车的起步价；
      N1←甲车的起步公里数；
      N2←乙车的起步公里数；
      X1←甲车起步价以后每公里的费用；
      X2←乙车起步价以后每公里的费用；
      n←欲行驶的里程数；
      if (n>=N1) P1←P1+X1*(n-N1+1)；
      if (n>=N2) P2←P2+X2*(n-N2+1)；
      if (P1<P2) 输出 (n," 公里时，乘甲车省钱")；
      else {
          if (P1>P2) 输出 (n," 公里时，乘乙车省钱")；
          else 输出 (n," 公里时，两车费用相同")；
      }
    
```



**知识回顾**

**程序设计三种基本结构**

在《信息技术基础》一书中，我们已经学习了程序设计的三种基本结构，即顺序结构、分支结构和循环结构。这三种结构虽然比较简单，但是可以用来描述出适用于计算机求解的各种算法。

三种结构的指令执行顺序有所不同。顺序结构就是按照指令出现的先后次序依次执行。条件指令是分支结构，它判断所给的条件是否成立，从而选择某一条路径的指令继续执行。循环指令是循环结构，它首先判断条件是否成立，如果不成立则直接跳出循环体，执行循环体外的第一条可执行指令；如果成立则执行循环体内的指令，然后再次判断条件是否成立，如果成立则再次执行循环体内的指令，如果不成立则跳出循环体。

(3) 流程图

流程图是用图表示算法的一种常用工具，其优点是直观易读。流程图中常用的符号如图 1-2-1 所示。

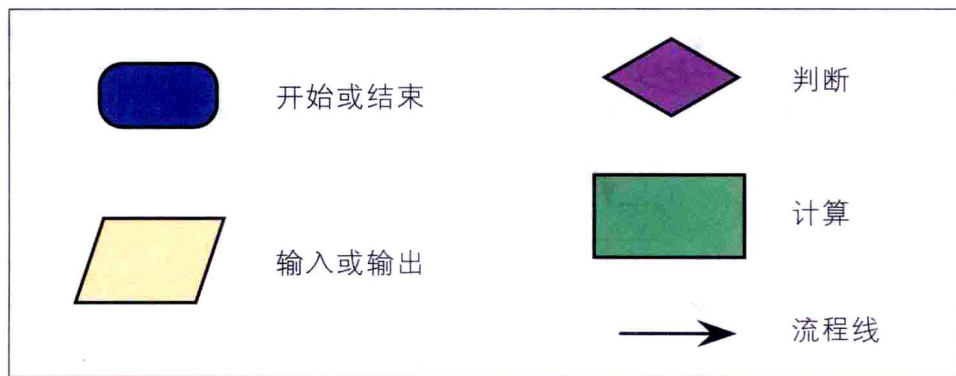


图 1-2-1 流程图的常用符号

用流程图表示的三种基本结构如图 1-2-2 所示。