高等院校“十二五”规划教材·数字媒体技术
示范性软件学院系列教材

C语言程序设计

丛书主编　肖刚强　　本书主编　刘月凡
副 主 编　戚海英 宋丽芳　　主　　审　李　瑞

辽宁科学技术出版社
LIAONING SCIENCE AND TECHNOLOGY PUBLISHING HOUSE

高等院校“十二五”规划教材·数字媒体技术
示范性软件学院系列教材

C语言程序设计

丛书主编　肖刚强
本书主编　刘月凡
副 主 编　戚海英　宋丽芳
主　　审　李　瑞

辽宁科学技术出版社
沈　阳

图书在版编目（CIP）数据

C语言程序设计 / 刘月凡主编. —沈阳：辽宁科学技术出版社，2012.2
高等院校“十二五”规划教材·数字媒体技术/肖刚强主编
ISBN 978-7-5381-7235-5

Ⅰ. ①C… Ⅱ. ①刘… Ⅲ. ①C语言—程序设计—高等学校—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2011）第239472号

出版发行：辽宁科学技术出版社
（地址：沈阳市和平区十一纬路29号　邮编：110003）
印 刷 者：辽宁美术印刷厂
经 销 者：各地新华书店
幅面尺寸：185mm × 260mm
印　　张：15.25
字　　数：370千字
印　　数：1~3000
出版时间：2012 年 2 月第 1 版
印刷时间：2012 年 2 月第 1 次印刷
责任编辑：于天文
封面设计：赵苗苗
版式设计：于　浪
责任校对：李淑敏

书　　号：ISBN 978-7-5381-7235-5
定　　价：32.00元

投稿热线：024-23284740
邮购热线：024-23284502
E-mail:lnkjc@126.com
http://www.lnkj.com.cn
本书网址：www.lnkj.cn/uri.sh/7235

序 言

当前，我国高等教育正面临着重大的改革。教育部提出的“以就业为导向”的指导思想，为我们研究人才培养的新模式提供了明确的目标和方向，强调以信息技术为手段，深化教学改革和人才培养模式改革，根据社会的实际需求，培养具有特色鲜明的人才，是我们面临的重大问题。我们认真领会和落实教育部指导思想后提出新的办学理念和培养目标。新的变化必然带来办学宗旨、教学内容、课程体系、教学方法等一系列的改革。为此，我们组织有多年教学经验的专业教师，多次进行探讨和论证，编写出这套“数字媒体技术”专业的系列教材。

本套系列教材贯彻了“理念创新，方法创新，特色创新，内容创新”四大原则，在教材的编写上进行了大胆的改革。教材主要针对软件学院数字媒体技术等相关专业的学生，包括了多媒体技术领域的多个专业方向，如图像处理、二维动画、多媒体技术、面向对象计算机语言等。教材层次分明，实践性强，采用案例教学，重点突出能力培养，使学生从中获得更接近社会需求的技能。

本套系列教材在原有学校使用教材的基础上，参考国内相关院校应用多年的教材内容，结合当前学校教学的实际情况，有取舍地改编和扩充了原教材的内容，使教材更符合当前学生的特点，具有更好的实用性和扩展性。

本套教材可作为高等院校数字媒体技术等相关专业学生的教材使用，也是广大技术人员自学不可缺少的参考书之一。

我们恳切的希望，大家在使用教材的过程中，及时给我们提出批评和改进意见，以利于今后我们教材的修改工作。我们相信经过大家的共同努力，这套教材一定能成为特色鲜明、学生喜爱的优秀教材。

肖刚强

前 言

计算机程序设计基础是一门十分重要的基础课程，是大学生学习计算机程序设计的入门课程，由于该课开设久远，在开设之初，一直沿用一种传统的理论研究式的教学模式，过于注重计算机语言的语法、语句格式的讲解，没有把计算机语言本身的目标是编程的逻辑思想放在主体地位上，对学生的编程能力训练不够，这样给后续课程的学习和研究留下了隐患。很多学生在学习这门课时感到枯燥难学，学过之后又不能用来解决实际问题。

我们作为从事计算机基础教学多年的教学团队，通过一线教学工作者长期的教学研究和总结经验，通过参加有关计算机基础教学研究会议，和其他高校从事计算机基础教学的同行们交流，大家都感到有必要改变我们的课程教学模式，用新的教学理念和方法培养新时代人才。我们开始研究对C语言程序设计课程的教学模式进行改革，以强调动手实践上机编程为切入点，以任务驱动方式，通过实例讲授程序设计的基本概念和基本方法，重点放在学习编程思路上，要求学生养成良好的编程习惯，在教学过程中注意培养学生的计算机语言的思维能力和编程动手能力，鼓励学生探索、研究和创新。根据多年的教学经验并结合学生的特点和需求，编写了《C语言程序设计》教材。该教材主要讲述了C语言程序设计的基础编程相关知识，同时在最后一章简单介绍了C语言综合设计。

本书由浅入深地介绍了C语言程序设计的基本思想、方法和步骤等，讲授的内容少而精，通过大量知识点明确的例题，让读者更好地掌握程序设计方法，强调从实践中学习，每章都附有大量的应用实例、习题以及近年来的二级C语言国家等级考试的试题解析。

本书共分11章，深入浅出地介绍了C语言的基础知识和程序设计思想，并力求达到在设计过程中来学习程序设计语言。本书由大连交通大学的刘月凡、戚海英、宋丽芳编写，李瑞老师主审，其中第1~6章由刘月凡编写，第7~11章由戚海英编写，第8章由大连科技学院的宋丽芳编写。

本书在编写过程中力求符号统一，图表准确，语言通俗，结构清晰。

本书既可以作为高等院校计算机专业本科、软件工程专业以及数字媒体技

术专业学生学习计算机语言的入门教材，也是广大工程技术人员自学不可缺少的参考书之一。

由于作者水平有限，书中难免有疏漏之处，欢迎广大读者批评指正。

如需本书课件和习题答案，请来信索取，地址：mozi4888@126.com

作者

目 录

第1章 程序设计思想

程序设计通俗地说就是完成一件事情时对步骤的安排。而计算机程序设计则是指在计算机上完成一件事情的过程。是指要解决的一个任务，要完成的一件事情。也就是说，计算机程序设计：就是通过计算机解决问题的过程。这里面实际上有两个层面的问题，首先是解决问题的方法和步骤，其次是如何把解决问题的方法和步骤通过计算机实现。要想在计算机完成这个任务，得用计算机语言来完成，就如同和英国人说话要用英语，和日本人说话要用日语一样，和计算机说话要用计算机语言。

1.1 程序设计

程序设计（Programming）是指设计、编制、调试程序的方法和过程。上面我们已经说过，对于初学者，了解程序设计可以把解决问题的方法与步骤和在计算机上实现这个过程分开来考虑。解决问题的方法与步骤，就是我们所说的算法。我们把算法在计算机上实现，也就完成了程序设计的过程。从这个过程来看，算法是程序的核心，是程序设计要完成的任务的灵魂。

1.1.1 程序设计的基本步骤

程序是指可以被计算机连续执行的多条指令的集合，也可以说是人与计算机进行“对话”的语言集合。人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中，当人为地给出命令之后，它就被计算机按指令操作顺序自动运行。

程序设计就是程序员设计程序的过程。

广义上说，程序设计就是用计算机解决一个实际应用问题时的整个处理过程，包括提出问题、确定数据结构、确定算法、编程、调试程序及书写文档等。

具体分析如下：

（1）提出问题：提出需要解决的问题，形成一个需求任务。

（2）确定数据结构：根据需求任务提出的要求，指定的输入数据和输出结果，确定存放数据的数据结构。

（3）确定算法：针对存放数据的数据结构确定问题、实现目标的步骤。

（4）编写程序：根据指定的数据结构和算法，使用某种计算机语言编写程序代码，输入到计算机中并保存在磁盘上，简称编程。

（5）调试程序：消除由于疏忽而引起的语法错误或逻辑错误；用各种可能的输入数据对程序进行测试，使之对各种合理的数据都能得到正确的结果，对不合理的数据都能进行适当处理。

（6）书写文档：整理并写出文档资料。

在程序设计的过程中，“确定算法”是一个相当重要的步骤。广义上讲，算法是为了解决一个问题而采取的方法和步骤。例如，描述跆拳道动作的图解，就是“跆拳道的算法”；一首歌曲的乐谱也可以称为该歌曲的算法。计算机科学中的算法是指为解决某个特定问题而采取的确定且有限的步骤，它是为了解决“做什么”和“怎么做”的问题。可以

说，算法是程序设计的灵魂。著名科学家沃思（Nikiklaus Wirth）提出一个公式：

数据结构+算法=程序

其中，数据结构是对数据的描述，也就是程序数据的类型和组织形式，而算法则是对操作步骤的描述，是我们使用程序设计语言来解决问题的“思路”。一个算法应该具有以下几个特点：有穷性、确定性、可行性、有零个或多个输入、有一个或多个输出。

1.1.2 程序设计的学习方法

从程序设计的基本步骤上可以看出，要想学好程序设计，首先要了解和掌握算法的概念，然后再学习一门计算机语言，这样，才初步可以在计算机上进行程序设计的工作。本章主要介绍算法的概念和思想。从第2章开始，我们要详细学习C语言（计算机语言），通过学习并使用C语言来完成计算机程序设计工作，我们学习计算机语言的目的最终是要进行程序设计，学习计算机语言的语法、规则的目的是为了更好地掌握计算机语言。

目前的计算机语言已经从低级语言发展成为高级语言了，高级语言更方便用户使用，它的源代码都是文本型的。但是，计算机本身只能接受二进制编码的程序，它不能直接运行这种文本型的代码，需要通过一个翻译把高级语言远程序代码转换成计算机能识别的二进制代码，这样计算机才能执行。而这个翻译，在这里我们把它叫做编译系统，也可以看成是计算机语言的编程界面。

在这里，我们先介绍一下算法的概念和思想，然后再介绍一下计算机语言的上机环境，也就是C语言的编译系统。目前大家比较喜欢使用的C语言编译系统有turbo C和VC++环境。Turbo C简单灵活，适合初学者掌握；VC++是windows系统下的编程环境，界面友好。

1.2 算法

算法是解决问题的方法与步骤，比人们平时理解的数学中算法的概念要广义一些。算法是程序的核心，是程序设计要完成的任务的灵魂。不论是简单还是复杂的程序，都是由算法组成。算法不仅构成了程序运行的要素，更是推动程序正确运行、实现程序设计目的的关键。

1.2.1 算法概念

当我们要买东西时，就会先有目标，然后到合适的商店挑选想要的物品，然后结账，拿发票（收据）、离开商店；当要理发时，就会先到一家理发店，与理发师商量好发型、理发、结账；当要使用计算机时，就会先打开屏幕、开机、输入密码，然后使用。不论我们做什么事情，都有一定的步骤。算法（Algorithm）简单来说就是解题的步骤，我们可以把算法定义成解决某一确定类型问题的任意一种特殊的方法。算法是程序设计的“灵魂”，它独立于任何具体的程序设计语言，一个算法可以用多种编程语言来实现。算法是一组有穷的规则，它们规定了解决某一特定类型问题的一系列运算，是对解题方案的准确与完整的描述。在程序设计中，算法要用计算机算法语言描述出来，算法代表用计算机解一类问题的精确、有效的方法。

【例1-1】输入三个互不相同的数，求其中的最小值（min）。

首先设置一个变量min，用于存放最小值。当输入a，b，c三个不相同的数后，先将a与b进行比较，把相对小的数放入min，再把c与min 进行比较，若c小于min，则将c的数值

放入min替换min中的原值；若c大于min，则min值保持不变，最后min中就是三个数中的最小值。详细步骤如下：

（1）先将a与b进行比较，若a<b，则a→min，否则，b→min。

（2）再将c与min进行比较，若c<min，则c→min。

这样，min 中存放的就是三个数中的最小数。

求解一个给定的可计算或可解的问题，不同的人可以编写出不同的算法来解决同一个问题。例如，计算1999 + 2999 + 3999 + … + 9999．也许有的人会选择一个一个加起来，当然也有人会选择（2000−1）+（3000−1）+…+（10000−1）的算法。理论上，不论有几种算法，只要逻辑正确并能够得出正确的结论就可以，但是，为了节约时间、运算资源等，我们当然提倡简单易行的算法。制订一个算法，一般要经过设计、确认、分析、编码、测试、调试、计时等阶段。

对算法的研究主要应包括5个方面的内容：

（1）设计算法。算法设计工作的完全自动化是不现实的，算法的设计最终还是要我们自己来完成，应学习和了解已经被实践证明可行的一些基本的算法设计方法，这些基本的设计方法不仅适用于计算机科学，而且适用于电气工程、运筹学等任何与算法相关的其他领域。

（2）表示算法。算法的类型不同，解决的问题不同，解决问题的步骤不同，表示算法的方法也自然有很多种形式，例如自然语言、图形、算法语言等。这些表示方式各有特色，也分别有适用的环境和特点。

（3）认证算法。算法认证其实就是确认这种算法能够正确地工作，达到解决问题的目的，即确认该算法具有可行性。正确的算法用计算机算法语言描述，构成计算机程序，计算机程序在计算机上运行，得到算法运算的结果。

（4）分析算法。对算法进行分析，确认这个算法解决问题所需要的时间和存储空间，并对其进行定量分析。对一个算法的分析可以很好地预测一种算法适合的运行环境，从而判断出其适合解决的问题。

（5）验证算法。用计算机语言将算法描述出来，进行运行、测试、调试，客观地判断算法的实际应用性、合理性。

1.2.2 算法的特性

一个算法应当具有以下5方面特性：

（1）确定性。与我们日常的行为不同，算法绝对不能有含糊其辞的步骤，如“请把那天的书带来！”这种无法明确哪一天、哪一本书、带到哪里的语句是不能够出现在算法中的，否则，算法的运行将变得无所适从。算法的每一步都应当是意义明确、毫不模糊的。

（2）可行性。算法的基本目的是解决问题，所以要求算法至少是可以运行并能够得到确定的结果的，不能存在违反基本逻辑的步骤。

（3）输入。一个算法有0个或多个输入，在算法运算开始之前给出算法所需数据的初值，这些输入取自特定的对象集合。

（4）输出。作为算法运算的结果，一个算法产生一个或多个输出，输出是同输入有某种特定关系的量。

（5）有穷性。一个算法应包含有限个操作步骤，而不能是无限的。一个算法总是在执行了有穷步的运算后终止，即该算法是可达的。

满足前4个特性的一组规则不能称为算法，只能称为计算过程，操作系统是计算过程的一个例子，操作系统用来管理计算机资源，控制作业的运行，没有作业运行时，计算过程并不停止，而是处于等待状态。

1.2.3 算法的表示

由于算法的步骤繁简不同，解决的问题不同，算法的表示方法也有许多种，一般可以归纳为以下几种。

1. 自然语言表示

自然语言，简单来说就是我们通常日常生活中应用的语言。相对于计算机语言来说，自然语言更容易被接受，也更容易学习和表达，但是自然语言往往冗长繁琐，而且容易产生歧义。例如，“他看到我很高兴。”便不清楚是他高兴，还是我高兴。尤其是在描述分支、循环算法时，用自然语言十分不方便。所以，除了一些十分简单的算法外，我们一般不采用自然语言来表示算法。

2. 图形表示

用图形表示算法即用一些有特殊意义的几何图形来表示算法的各个步骤和功能。使用图形表示算法的思路是一种很好的方法，图形的表示方法比较直观、清晰，易于掌握，有利于检查程序错误，在表达上也克服了产生歧义的可能。一般我们使用得比较多的有传统流程图、N—S流程图、PAD流程图等。本书只介绍传统流程图，其他流程图请参看其他程序设计书籍。

传统的流程图一般由图1-1所示的几种基本图形组成。

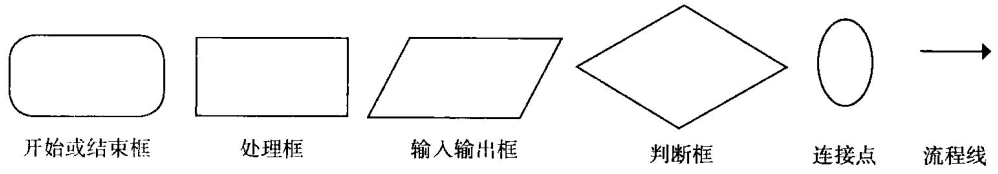

图1-1 流程图的基本图形

【例1-2】输入两个整数给变量x和y，交换x和y的值后再输出x和y的值。

分析：完成本题需要3个步骤，首先输入两个整数给x和y；之后交换x和y的值；最后输出x和y。

根据以上分析，容易画出程序的流程图（图1-2），并根据流程图写出程序：

```
main()
{   int w,x,y;
    printf("请输入两个整数：");
    scanf("%d%d",&x,&y);
    w=x;
    x=y;
    y=w;
    printf("交换后：x=%dy=%d\n",x,y);
}
```

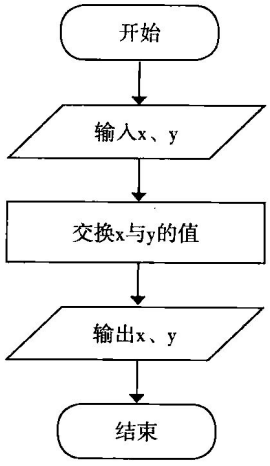

图1-2 流程图

说明：

（1）scanf函数为输入函数，可以用来输入数据；输出数据可以使用printf函数。

（2）引入第3个变量w，先把变量x的值赋给变量w，再把变量y的值赋给x，最后把变量w的值赋给y，最终达到交换变量x和y的值的目的。引入w的作用是交换变量x和y的值。交换x和y的值不能简单地用“x=y;”和“y=x;”这两个语句，如果没有把x的值保存到其他变量，就执行“x=y;”语句，把y的值赋给x，将使x和y具有相同的值，丢失x原来的值，也就无法实现两个值的交换。

假如输入3和9，运行程序时，屏幕可能显示如下信息：

请输入两个整数：3 9

交换后：x=9 y=3

【例1-3】输入a，b，c三个数，把最小的值输出，流程图如图1-3所示。

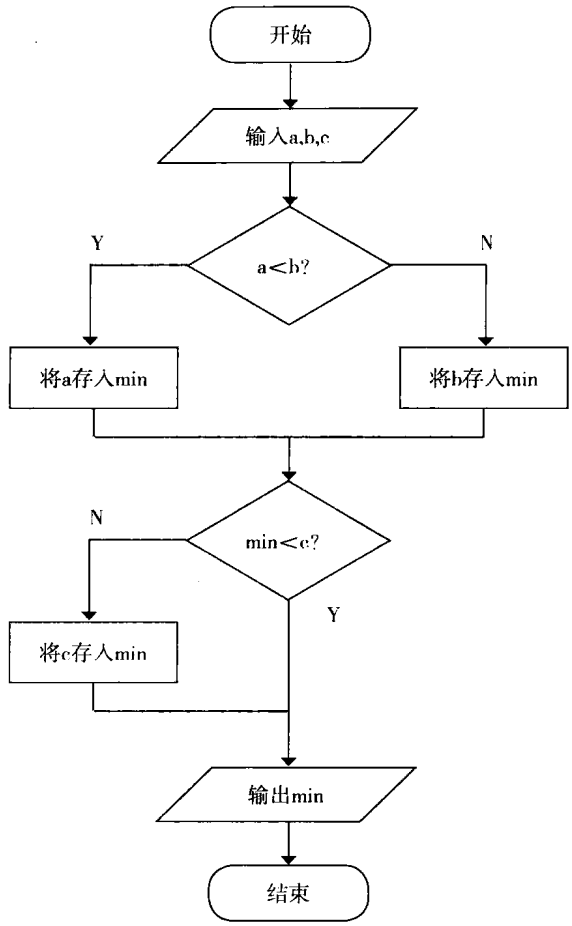

图1-3　流程图

先将a与b进行比较，若a<b，则将a存入min，否则，将b存入min；再将c与min进行比较，若c<min，则将c存入min，然后输出min，否则，直接输出min。

根据以上几个例子可以看出，使用传统的流程图主要由带箭头的线、文字说明和不同形状的框构成。采用传统的流程图可以清晰直观地反映出整个算法的步骤和每一步的先后顺序。因此，相当长的一段时间内，传统流程图成为很流行的一种算法描述方式。

但是当算法相当复杂、篇幅很长时，使用传统的流程图就会显得既费时又费力。随着结构化程序设计思想的推行与发展，渐渐地衍生出N—S结构化流程图。

3. 伪代码

伪代码（Pseudocode）是一种算法描述语言。使用伪代码的目的是为了使被描述的算法可以较容易地以任何一种编程语言（Pascal, C, Java, etc）实现。因此，伪代码必须结构清晰，代码简单，可读性好，并且类似自然语言。

用各种算法描述方法所描述的同一算法，只要该算法的功用不便，就允许在算法的描述和实现方法上有所不同。

1.2.4 算法的复杂度

找到求解一个问题的算法后，接着就是该算法的实现，至于是否可以找到实现的方法，取决于算法的可计算性和计算的复杂度，该问题是否存在求解算法，能否提供算法所需要的时间资源和空间资源。

算法的复杂度是对算法运算所需时间和空间的一种度量。在评价算法性能时，复杂度是一个重要的依据。算法复杂度的程度与运行该算法所需要的计算机资源的多少有关，所需要的资源越多，表明该算法的复杂度越高；所需要的资源越少，表明该算法的复杂度越低。计算机的资源，最重要的是运算所需的时间和存储程序和数据所需的空间资源，算法的复杂度有时间复杂度和空间复杂度之分。

对于任意给定的问题，设计出复杂度尽可能低的算法是在设计算法时考虑的一个重要目标。另外，当给定的问题已有多种算法时，选择其中复杂度最低者，是在选用算法时应遵循的一个重要准则。因此，算法的复杂度分析对算法的设计或选用有着重要的指导意义和实用价值。

算法的时间复杂度是指算法需要消耗的时间资源。因此，问题的规模n 越大，算法执行的时间的增长率与相应的复杂度函数的增长率正相关。

算法的空间复杂度是指算法需要消耗的空间资源。其计算和表示方法与时间复杂度类似，一般都用复杂度的渐近性来表示。同时间复杂度相比，空间复杂度的分析要简单得多。

1.2.5 结构化程序设计方法

程序设计初期，由于计算机硬件条件的限制，运算速度与存储空间都迫使程序员追求高效率，编写程序成为一种技巧与艺术，而程序的可理解性、可扩充性等因素被放到第二位。这一时期的高级语言FORTRAN、COBOL、ALGOL、BASIC等，由于追求程序的高效率，不太注重所编写程序的结构。存在的一个典型问题就是程序中的控制随意跳转，即不加限制地使用goto语句。goto语句允许程序从一个地方直接跳转到另一个地方去。执行这种语句的好处是程序设计十分方便灵活，减少了人工复杂度，但其缺点是一大堆跳转语句使得程序的流程十分复杂紊乱，难以看懂也难以验证程序的正确性，如果有错，排起错来更是十分困难。这种转来转去的流程图所表达的混乱与复杂，正是软件危机中程序人员处境的一个生动写照。而结构化程序设计，就是要把这团乱麻理清。

经过研究，人们发现任何复杂的算法都可以由顺序结构、选择（分支）结构和循环结构这三种基本结构组成。因此，我们构造一个算法的时候，也仅以这三种基本结构作为“建筑单元”，遵守三种基本结构的规范，基本结构之间可以并列，可以相互包含，但不允许交叉，不允许从一个结构直接转到另一个结构的内部去。正因为整个算法都是由三种基本结构组成的，就像用模块构建的一样，所以结构清晰，易于正确性验证，易于纠错，这种方法，就是结构化方法。遵循这种方法的程序设计，就是结构化程序设计。C语言就是一种结构化语言。

1. 顺序结构

顺序结构表示程序中的各操作是按照它们出现的先后顺序执行的，其流程如图1-4所示。整个顺序结构只有一个入口点和一个出口点。这种结构的特点是：程序从入口点开始，按顺序执行所有操作，直到出口点处，所以称为顺序结构。程序的总流程都是顺序结构的。

2. 选择结构

选择结构表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择三种形式。

双选择是典型的选择结构形式，其流程如图1-5所示，在这两个分支中只能选择一条且必须选择一条执行，但不论选择了哪一条分支执行，最后流程都一定到达结构的出

口点处。

多选择结构是指程序流程中遇到多个分支，程序执行方向将根据条件确定。如果满足条件1则执行S1处理，如果满足条件n则执行Sn处理，总之要根据判断条件选择多个分支的其中之一执行。不论选择了哪一条分支，最后流程要到达同一个出口处。如果所有分支的条件都不满足，则直接到达出口。

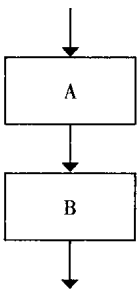

图 1-4　顺序结构

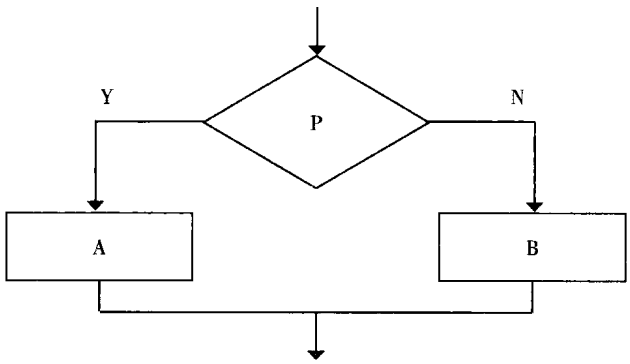

图 1-5　选择结构

3. 循环结构

循环结构表示程序反复执行某个或某些操作，直到某条件为假（或为真）时才可终止循环。在循环结构中最主要的是：什么情况下执行循环？哪些操作需要循环执行？循环结构的基本形式有两种：当型循环和直到型循环，其流程如图1-6（a）（b）所示。图中A的操作称为循环体，是指从循环入口点到循环出口点之间的处理步骤，这就是需要循环执行的部分。而什么情况下执行循环则要根据条件判断。

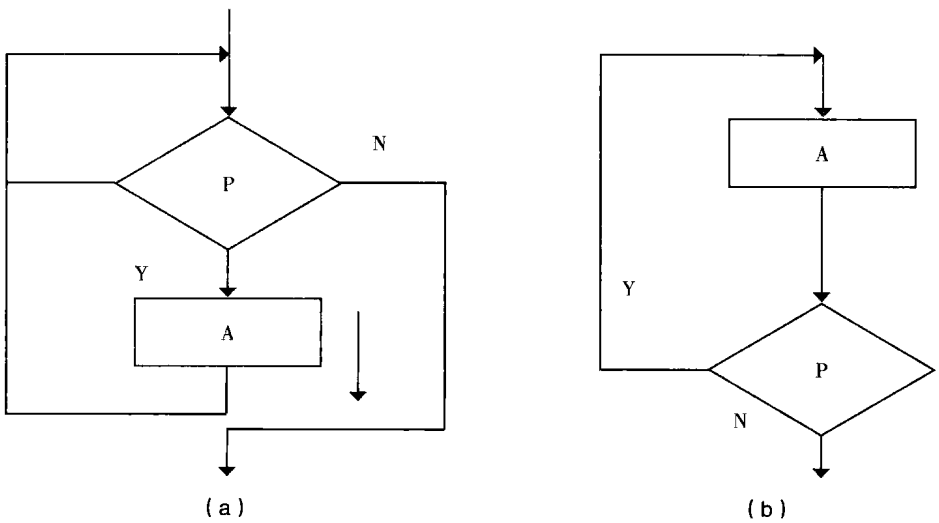

图1-6　循环结构

当型循环：表示先判断条件，当满足给定的条件时执行循环体，并且在循环终端处流程自动返回到循环入口；如果条件不满足，则退出循环体直接到达流程出口处。因为是