第 3 章：程序语言和语言处理程序基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

考点脉络

程序语言和编译相关知识，主要在编码阶段应用。在现实生活中，软件设计师往往不仅仅只做纯设计工作，还需要完成一些编码工作，所以要求掌握程序语言相关知识。

根据考试大纲，本章要求考生掌握以下几个方面的知识点。

（1）汇编、编译、解释系统的基础知识和基本工作原理。

（2）程序设计语言的基本成分：数据、运算、控制和传输，程序调用的实现机制。

（3）各类程序设计语言的主要特点和适用情况：过程式程序语言、面向对象程序设计语言、函数式程序设计语言、逻辑程序设计语言的基本特点、脚本语言的特点。

从历年的考试情况来看，本章的考点主要集中以下方面。

在汇编、编译、解释系统的基础知识和基本工作原理中，主要考查文法、有限自动机、正规式的相关内容。

在程序设计语言的基本成分中，主要考查语句的作用、语句的语义、程序的控制结构、函数调用的参数传递。

在各类程序设计语言的主要特点和适用情况中，主要考查各种程序语言的特点比较。

上一节　本书简介　下一节

第 3 章：程序语言和语言处理程序基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

汇编、编译、解释系统基础

本节将针对软件设计师考试当中需要掌握的编译原理知识进行描述。编译原理是计算机专业课程中最难的课程之一，所以考生在学习相关内容时遇到问题需要冷静思考，切忌急于求成。同时该考点近年考查分值也逐步减少，所以本节内容将做大量裁剪，考生只需掌握有限的几个知识点。

上一节　本书简介　下一节

第 3 章：程序语言和语言处理程序基础知识　　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

考点精讲

1. 解释与编译

在计算机中，程序语言分为低级语言和高级语言，如汇编语言便是一种低级语言，而平时我们所使用的：Java、C#、Delphi等，都属于高级语言。使用高级语言开发的程序是不能直接运行的。

需要经过一系列的处理，才能运行。这个过程，根据其处理方式的不同，可分为：解释型和编译型。

解释型：接受所输入的用程序语言编写的源程序，然后直接解释执行；这类语言中的最典型代表是BASIC。

编译型：它是将用某种程序语言编写的源程序直接翻译成为另一种语言（目标语言程序），而且两者在逻辑上等价。如：C语言。

2. 编译过程

所谓编译过程，就是使用编译程序将高级语言源程序翻译为等价的机器语言程序的过程。一般来说，编译程序分为以下几个部分：词法分析、语法分析、语义分析、中间代码生成、代码优化、目标代码生成以及贯穿始终的表格管理与出错处理。各部分之间的关系如图3-1所示。

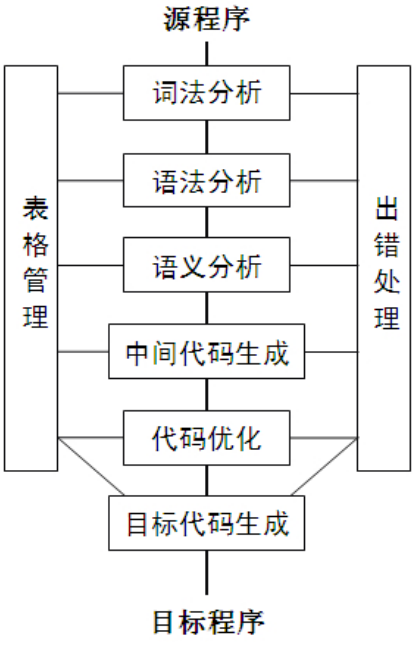


图3-1编译的主要过程

3.语言及文法的概念

语言是按照一定规则排列的符号和集合。要形式化地描述一个语言，就需要借助文法的概念。文法就是用来描述语言的语法结构的形式规则。

根据乔姆斯基的分类法，文法可以分成四种类型，如表3-1所示。

表3-1乔姆斯基分类法

类型	别称	说明	对应自动机
0 型	短语文法	G 的每条产生式α→β满足 a 属于 V 的正则闭包，且至少含有一个非终结符，而β属于 V 的闭包	图灵机
1 型	上下文有关文法	G 的任何产生式α→β满足|α|<=|β|，仅仅 S→ε例外，但 S 不得出现在任何产生式右部	线性界限自动机
2 型	上下文无关文法	G 的任何产生式为 A→β，A 为非终结符，B 为 V 的闭包	非确定的下推自动机
3 型	正规文法	G 的任何产生式为 A→αB 或 A→α，α属于非终结符的闭包，A、B 都属于非终结符	有限自动机

要根据这样的定义来对文法进行判断，总是让许多考生无从下手，其实只要掌握一些规律，就能够更好地完成这一任务（当然，这个知识点考查概率在不断降低，若完成本节学习，仍有疑问，可考虑先将主要精力投入其它章节学习）。

0型文法：就是一般形式的方法，只要符合前面的定义，就属于0型文法。因此也称为无限制文法、短语文法。由0型文法生成的语言称为0型语言。

1型文法：它的每一个产生式应该满足以下条件：$\varphi_1 A\phi \rightarrow \varphi_1 a\phi$；其中：$A \in V$（也就是说，A属于非终结符），$\phi$，$\varphi_1$，$\alpha \in (V \cup T)^*$（也就是说，$\phi$，$\varphi_1$，$\alpha$是非终结符或终结符的组合）。由于这些产生式在替换非终结符时，需要考虑它的前一个、后一个元素（也就是产生式替换的非终结符的前、后均有一个不变的符号），因此又称为上下文有关文法，其产生的就是上下文有关语言，即1型文法。

2型文法：如果它的所有产生式都形如：$A \rightarrow \alpha$；其中：$A \in V$（也就是说，A属于非终结符），$\alpha \in (V \cup T)^*$ (也就是说，A是非终结符和/或终结符的组合）。那么，它就是一个2型方法，由于转换与前后元素没有关系（也就是产生式的前后都是空的），因此，也称之为上下文无关文法。

3型文法：它也是一个2型方法。

右线性文法：如果它的所有产生式都形如：$A \rightarrow \alpha$ B或$A \rightarrow \alpha$（也就是除了$A \rightarrow \alpha$这种特殊形式外，每个产生式的右边都有一个非终结符）；其中：A，$B \in V$（也就是说，A，B属于非终结符），$\alpha \in (V \cup T)^*$ (也就是说，α是非终结符和/或终结符的组合)。

左线性文法：如果它的所有产生式都形如：$A \rightarrow B\alpha$ 或$A \rightarrow \alpha$（也就是除了$A \rightarrow \alpha$这种特殊形式外，每个产生式的左边都有一个非终结符）；其他与上面一样。

右线性、左线性都称做3型文法或正则文法。

4. 词法分析

词法分析是整个分析过程的一个子任务，它把构成源程序的字符串转换成语义上关联的单词符号（包括关键字、标识符、常数、运算符和分界符等）的序列。词法分析可以借助于有限自动机的理论与方法进行有效的处理。

（1）有限自动机

有限自动机是一种自动识别装置，能够准确地识别正规集。它与3型文法对应，可以分为确定的有限自动机和不确定的有限自动机两种。

确定的有限自动机（DFA）

M=(S, Σ，δ,S0,Z)

1）S是一个有限集，每个元素为一个状态

2）Σ 是一个有穷字母表，每个元素为一个输入字符

3）δ是转换函数：是一个单值对照

4）S0,属于S，是其唯一的初态

5）Z是一个终态集（可空）

有限状态自动机可以形象地用状态转换图表示，设有限状态自动机：

DFA = ({S, A, B, C, f}, {1, 0},δ,S, {f})，

其中：

δ(S, 0) = B, δ(S, 1) = A, δ(A, 0) = f, δ(A, 1) = C, δ(B, 0) = C, δ(B, 1) = f,δ(C, 0) = f, δ(C, 1) = f

其对应的状态转换图如图3-2所示。

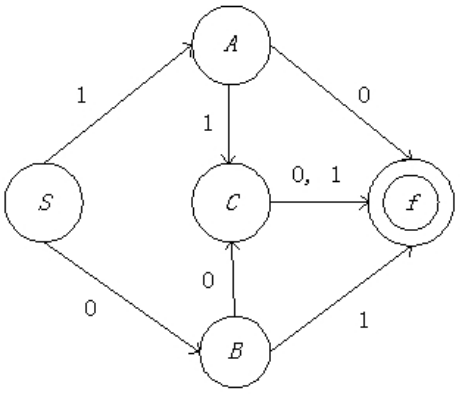


图3-2状态转换图

不确定的有限自动机（NFA）

M=(S, Σ ，δ,S0,Z)

1）S是一个有限集，每个元素为一个状态

2） Σ 是一个有穷字母表，每个元素为一个输入字符

3）δ是转换函数，是多值对照

4）S0,属于S，是非空初态集

5）Z是一个终态集（可空）

与确定的有限自动机一样，不确定有限自动机同样可以用状态转换图表示，所不同的是，在图中一个状态结点可能有一条以上的边到达其它状态结点。

从定义的角度来区分NFA与DFA，我们发现：DFA是单值对照，NFA是多值对照（其实也就对应着上面所述的“NFA图中一个状态结点可能有一条以上的边到达其它状态结点”）。

NFA可以转换为等价的DFA。

（2）正规式

对于字母表 Σ 而言，正规则式和它所表示的正规集的递归定义是：

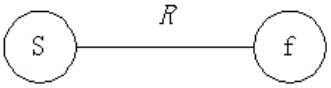
空集是正规表达式。

任何属于 Σ 的 α ，都是其正规式。

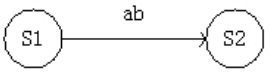
假定U和V都是 Σ 上的正规式，那它们的或、连接、闭包都是正规式。

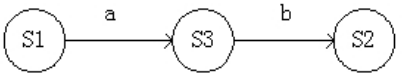
通常在正规表达式中，一元运算符“*”具有最高的优先级，连接运算具有次优先级，运算符“|”具有最低优先级，这三个运算都是左结合的。每一个正规表达式R都对应一个有限自动机M，使M所接受的语言就是正规表达式的值。经过以下步骤可以从一个正规表达式R构造出相应的有限自动机M。

首先定义初始状态S和终止状态f，并且组成有向图：

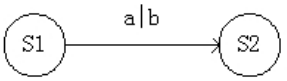


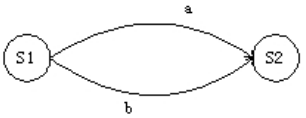
然后反复应用以下规则：

若 

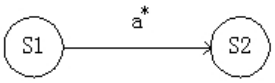
，则用 

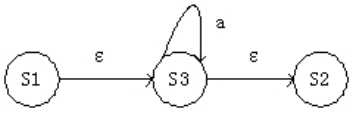
代替；

若 

，则用 

代替；

若 

，则用 

代替；

直到所有的边都以Σ中的字母或ε标记为止。由此产生了一个带ε–转移的非确定有限自动机，然后可以通过上面介绍的方法，把该自动机转换成确定有限状态自动机。

下面举一个例子说明自动机理论在词法分析程序中的应用。C语言中对标识符的规定为由“_”或以字母开头的由“_”、字母和数字组成的字符串，该标识符的定义可以表示为下面的正则表达式：

$$(\_ \mid a)(\_ \mid a \mid d)^*$$

式中的α代表字母字符{A,…,Z,a,…,z}，d代表数字字符{0,1,…,9}。利用前面的方法构造出如图3-3所示的有限自动机。

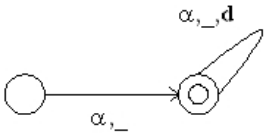


图3-3有限自动机图例

该自动机所接受的语言就是C语言中的标识符。

在有限自动机的状态转换过程中，需要执行相关的语义动作。例如当识别到一个标识符时，需要在符号表中添加该标识符，并且向语法分析程序输送表示该标识符的单词。

5. 语法分析

语法分析的任务是识别由词法分析给出的单词符号序列是否为给定文法的正确句子（程序），语法分析可以分为自底向上分析和自顶向下分析两大类。

（1）自底向上分析

自底向上分析也称为移进——归约分析法。它的基本思想是：对输入符号串自左向右进行扫描，并将输入符号逐一移进一个后进先出的栈中，边移进边分析；一旦栈顶符号串形成某个句型的可归约串时，就用某产生式的左部非终结符来替代（这称之为归约）。一直重复这个过程，直到栈中只剩下文法的开始符号。

（2）自顶向下分析

自顶向下分析法也称为面向目标的分析方法，也就是从文法的开始符号出发尝试推导出与输入的单词串相匹配的句子。

上一节　本书简介　下一节

第 3 章：程序语言和语言处理程序基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

一点一练

试题1

如图3-4所示为一个有限自动机（其中，A是初态、C是终态），该自动机所识别的字符串的特

点是__(1)__。

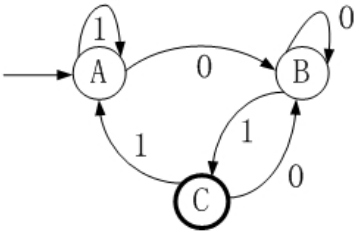


图3-4有限自动机图例

（1）A．必须以11结尾的0、1串 B．必须以00结尾的0、1串

C．必须以01结尾的0、1串 D．必须以10结尾韵0、1串

试题2

编译和解释是实现高级程序设计语言翻译的两种基本形式。以下关于编译与解释的叙述中，正确的是__(2)__。

（2）A．在解释方式下，对源程序不进行词法分析和语法分析，直接进行语义分析

B．在解释方式下，无需进行语法、语法和语义分析，而是直接产生源程序的目标代码

C．在编译方式下，必须进行词法、语法和语义分析，然后再产生源程序的目标代码

D．在编译方式下，必须先形成源程序的中间代码，然后再产生与机器对应的目标代码

试题3

若C程序的表达式中引用了未赋初值的变量，则__(3)__。

（3）A．编译时一定会报告错误信息，该程序不能运行

B．可以通过编译并运行，但运行时一定会报告异常

C．可以通过编译，但链接时一定会报告错误信息而不能运行

D．可以通过编译并运行，但运行结果不一定是期望的结果

试题4

以下关于高级程序设计语言翻译的叙述中，正确的是__(4)__。

（4）A．可以先进行语法分析，再进行词法分析

B．在语法分析阶段可以发现程序中的所有错误

C．语义分析阶段的工作与目标机器的体系结构密切相关

D．目标代码生成阶段的工作与目标机器的体系结构密切相关

试题5

下图所示为一个有限自动机（其中，A是初态、C是终态），该自动机可识别__(5)__。

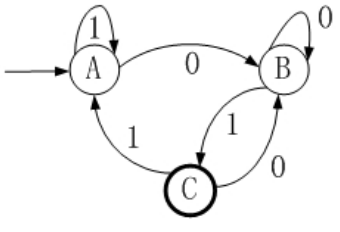


图3-5有限自动机图例

（5）A．0000　　B．1111　　C．0101　　D．1010

试题6

以下关于变量和常量和叙述中，错误的是__(6)__。

（6）A．变量的取值在程序运行过程中可以改变，常量则不行

B．变量具有类型属性，常量则没有

C．变量具有对应的存储单元，常量则没有

D．可以对变量赋值，不能对常量赋值

　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

解析与答案

试题1分析

被有限自动机所识别是指从初态开始到终态结束，所输入的字符串能够按顺序地执行下去，若到某个状态不能往下走得到下一个字符，则认为不能识别。

在本题中，从初态A出发，不管经过多少个1和0之后，只能是处在A、B、C三种状态中的一种，所以在(0|1)*后，只能是处在A、B、C三种状态中的一种，不管是在那个状态，输入0后，都会处在状态B，然后输入1，都会转换到状态C，因此与本题有限自动机等价的正规式是(0|1)*01，即该自动机所识别的字符串的特点是必须以01结尾的0、1串。

试题1答案

（1）C

试题2分析

编译和解释是语言处理的两种基本方式。编译过程包括词法分析、语法分析、语义分析、中间代码生成、代码优化和目标代码生成等阶段，以及符号表管理与出错处理模块。

解释过程在词法、语法和语义分析方面与编译程序的工作原理基本相同，但是在运行用户程序时，它直接执行源程序或源程序的内部形式。

这两种语言处理程序的根本区别是：在编译方式下，机器上运行的是与源码程序等价的目标程序，源程序和编译程序都不再参与目标程序的执行过程；而在解释方式下，解释程序和源程序（或其某种等价表示）要参与到程序的运行过程中，运行程序的控制权在解释程序。

在编译方式下，词法、语法和语义分析是必须要进行的工作，而生产中间代码和优化则是可以进行也可以不进行。

试题2答案

（2）C

试题3分析

在C程序中，若在某个表达式中引用了未赋初值的变量，那么程序是可以通过编译并运行的，因为程序中并没用语法方面的错误，只是运行的结果可能与我们期望的结果不一致。

试题3答案

（3）D

试题4分析

在对用高级程序设计语言编写的程序进行执行时，首先是将源代码翻译成目标代码，然后在连接成可执行的二进制代码。因此在翻译阶段，目标代码生成阶段的工作与目标机器的体系结构密切

相关。

试题4答案

（4）D

试题5分析

本题主要考查有限自动机。

在本题中，A是初始状态，C是终止状态，通过选项中的字符串可以从初始状态到达终止状态，则说明该字符串能被题目中的自动机识别。也可以理解为依次输入选项中的字符串，可以在该自动机中找到相应的路径。

对于选项A的字符串0000，在输入0后，从初始状态A转移到状态B，然后接着输入3个0，状态然后停留在B，而无法到达终态C，因此选项A不能被该自动机识别。

同样的道理，我们可以找到字符串0101能被该自动机识别，在输入0后，状态跳转到B，输入1则由B转至C，再输入0，又由C转至B，最后输入1，由B转至终态C。

试题5答案

（5）C

试题6分析

本题主要考查我们对常理与变量的理解。顾名思义，常理是指值一旦确定后就不能再变的量，而变量则是一个在程序执行过程中，可以根据需要修改的量，是一个可改变的量。当然不管是常理还是变量，它们都有其类型属性。

试题6答案

（6）B

上一节　本书简介　下一节

程序设计语言基础

程序设计语言基础主要包括程序当中的数据、运算、控制、传输以及程序调用的实现机制等相关内容。根据历年考试的出题情况，本节主要考查表达式的转化和程序调用的实现机制。

上一节　本书简介　下一节

考点精讲

1. 数据类型

计算机系统中，会把数据分为不同的类型，常见的数据类型包括：整型，长整型，字符型，浮

点型等。

数据类型的出现是为了把数据分成所需内存大小不同的数据，编程的时候需要用大数据的时候才需要申请大内存，就可以充分利用内存。例如大胖子必须睡双人床，就给他双人床，瘦的人单人床就够了。

要求源程序中的数据必须具有类型的目的主要有以下几个方面：

第一是方便为数据合理分配存储单元；第二是规定了数据类型，就知道了其占用的字节数，从而也就规定了数据对象的取值范围及能够进行的运算；第三是对参与表达式求值的数据对象可以进行合法性检查，比如浮点数就不能进行自加操作。

2. 表达式

表达式是程序语句的基本组成，体现了程序控制和数据改变的方法。表达式可分为前缀表达式、中缀表达式与后缀表达式。该知识点在多次考试中都考到了，是一个重点。

（1）中缀表达式

中缀表示法是日常最通用的用法，但是在程序语言中使用中缀表示法会产生下面的问题。

由于中缀表示法仅适合于二元操作符，一种语言不能只使用中缀表示法，而必须结合中缀与前缀表示法。这种混合使用会使翻译过程相对复杂。

当一个以上的中缀操作符出现在表达式中时，如果不使用括号就有可能产生二义性。

（2）前缀表达式（波兰式）

在前缀表达式中，是以从左到右的顺序先写操作符后写操作数，如果操作数本身是一个具有操作数的操作，则对其使用同样的规则。例如，公式(a+b)*(a-b)，使用前缀表达式表示将变为 * + a b – a b。函数调用可以看做一种前缀表达式，因为一般是把操作符函数名写在它的参数的左边，像f(a, b)这样。

以前缀表示的表达式可以在一次扫描后计算出值，前提是要明确知道每个操作符的参数的数目。用以下的算法通过使用一个执行栈，计算给定的前缀表达式P。

如果P的下一项是一个操作符，将它压入栈，并把参数计数器设置为该操作符所需要的参数的数目n。

如果P的下一项是一个操作数，把它压入栈。

如果栈顶操作数的个数为n，则把操作符作用于这n个操作数，得出的结果替换该操作符和它所有的参数，作为操作数压入栈。

前缀表达式的计算方法意味着在每个操作数压入栈后都必须检查操作数的数目是否满足最近栈顶的操作符的要求，而后缀表达式就无需做这种检查。

（3）后缀表达式（逆波兰式）

后缀表示法类似于前缀表示法，不同之处在于操作符跟在操作数之后。前面的公式使用后缀表示法时表示为a b + a b– * 。由于这一点不同，在计算后缀表达式的时候，当扫描到操作符时，栈中已压入了它的操作数。因此计算后缀表达式的算法如下。

如果P的下一项是操作数，将它压入栈。

如果P的下一项是n元操作符（即需要参数个数为n的操作符），那么它的参数就是栈顶的n项，把该操作符作用于这n项，得到的结果作为操作数替代栈顶的n项。

由于后缀表达式的计算是直接的，并且易于实现，因此它是很多翻译器产生表达式代码的基础。

在考试当中，该知识点的考查，通常形式是给出一个表达式的中缀表达形式（或前缀、后缀），让考生将其转换为前缀或后缀表达形式。

2.程序调用的实现机制

在过程式程序设计语言中，习惯将程序看做层次结构：从主程序开始执行，然后进入各层次的过程执行，最后返回主程序。而过程通常有四个要素，如图3-6所示。

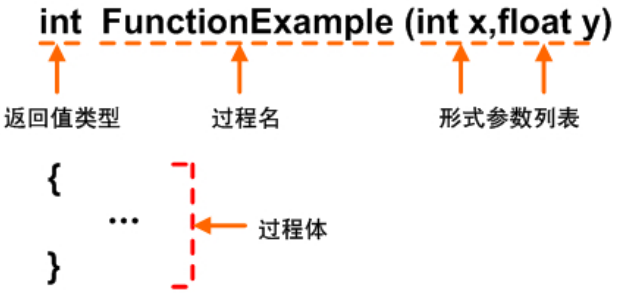


图3-6过程四要素

当用户调用一个过程时，就将通过参数来传递信息。形式参数就是过程定义中用于命名所传递的数据或其他信息的标识符（如图3-6所示）；而实际参数是在调用点表示向被调用过程传递的数据或其他信息的表达式。多数语言中，形参与实参的对应关系是按位置来进行的，因此在调用时，实参的个数、类型与顺序应该与形参一致。常见的参数传递方式有两种：传值与引用（传址）两种调用方式，如表3-2所示。

表3-2参数的传递方式

传递方式	主要特点
传值调用	形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变
引用（传址）调用	形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值

例如下面这个程序，如果分别假设采用传值调用和引用调用，那么B[1]和B[2]的值分别应该是多少？

```
program main（input，output）
   var  I:integer；
      B:array[1..2] of integer;
   Procedure  Q (x:integer)
   Begin
      I:=1;x:=x+2;B[I]:=10;
      I:=2;x:=x+2;
   End
   Begin
      B[1]:=1;B[2]:=2;I:=1;Q(B[I]);
      Writeln(B[1],B[2]);
    End
```

首先分析程序，主程序中定义了整型变量I和整型数组B，子程序Q只有一个局部变量，也就是形式参数x，另外，其中还引用了两个变量I和B，根据作用域原则，I和B在子程序中没有定义，顺序查找它的外层程序，即主程序，在主程序中，我们找到了I和B的定义，因此，子程序Q中引用的I和B实际上是主程序中所定义的变量。

当为传地址调用时，主程序中对x的引用能改变实参的值，在函数调用Q(B[I])之前，I的值为1，

B[I]的值为B[1]。进入子程序Q后，由于是传地址调用，将B[1]的地址传递个形式参数x，然后程序中对x的引用都是通过B[1]的地址访问到B[I]的值。则子程序中的语句：

I:=1;x:=x+2;B[I]:=10;

I:=2;x:=x+2;

实际上相当于：

I:=1; B[1]:= B[1]+2;B[I]:=10;

B[1]:= B[1]+2;

因此子程序调用返回后B[1]的值为12，B[2]的值仍然为2。

当为传值调用时，则子程序对形式参数x的引用不会改变实参的值，但子程序中对主程序定义的变量I和B的赋值还是可以改变它们的值，对x的赋值不会对主程序产生影响，所以子程序相当于执行了以下语句：

I:=1; B[I]:=10;

I:=2;

因此子程序调用返回后B[1]的值为10，B[2]的值仍然为2。

第 3 章：程序语言和语言处理程序基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

一点一练

试题1

某程序设计语言规定在源程序中的数据都必须具有类型，然而，__(1)__并不是做出此规定的理由。

（1）A．为数据合理分配存储单元

B．可以定义和使用动态数据结构

C．可以规定数据对象的取值范围及能够进行的运算

D．对参与表达式求值的数据对象可以进行合法性检查

试题2

算术表达式采用逆波兰式表示时不用括号，可以利用__(2)__进行求值。与逆波兰式ab-cd+*对应的中缀表达式是__(3)__。

（2）A．数组 B．栈 C．队列 D．散列表

（3）A．a-b+c*d　B．（a-b）*c+d　C．（a-b）*（c+d）D．a-b*c+d

试题3

函数（过程）调用时，常采用传值与传地址两种方式在实参与形参间传递信息。以下叙述中，正确的是__(4)__。

（4）A．在传值方式下，将形参的值传给实参，因此，形参必须是常量或变量

B．在传值方式下，将实参的值传给形参，因此，实参必须是常量或变量

C．在传地址方式下，将形参的地址传给实参，因此，形参必须有地址

D．在传地址方式下，将实参的地址传给形参，因此，实参必须有地址

试题4

函数t、f的定义如下所示，其中，a是整型全局变量。设调用函数t前a的值为5，则在函数t中以传值调用（call by value）方式调用函数f时，输出为__(5)__；在函数t中以引用调用（ call by reference）方式调用函数f时，输出为__(6)__。

```
t(): int x = f(a);
     print a+x;

f(int r): a = r+1; r = r*2
          return r;
```

（5）A．12　　B．16　　C．20　　D．24

（6）A．12　　B．16　　C．20　　D．24

解析与答案

试题1分析

请参看3.3.1节中关于数据类型的描述。

试题1答案

（1）B

试题2分析

逆波兰式也叫后缀表达式，即将运算符写在操作数之后的表达式，它不需使用括号，在将算术表达式转换为逆波兰式表示时，需要分配2个栈，一个作为临时存储运算符的栈S1（含一个结束符号），一个作为输入逆波兰式的栈S2（空栈）。

而逆波兰式ab-cd+*转换为中缀表达式的过程为：ab-cd+* =（ab-）*（cd+）=（a-b）*（cd+）=（a-b）*（c+d）。因此本题答案选C。

试题2答案

（2）B （3）C

试题3分析

形式参数就是过程定义中函数名后括号中所带的参数；实际参数是在调用点表示向被调用过程传递的数据。

在函数调用时，数据传递的方向是从实参到形参。只是采用传值传递方式时，传递的是数值，这个数值只要是确定的即可，可以是常理、变量或表达式等。而采用传址传递方式时，传递的是地址，因此实参必须有地址。

试题3答案

（4）D

试题4分析

传值调用中，形参取的是实参的值，形参的改变不会导致调用点所传的实参的值发生改变；而

引用（传址）调用中，形参取的是实参的地址，即相当于实参存储单元的地址引用，因此其值的改变同时就改变了实参的值。

在本题中，首先是采用传值调用，这个时候将变量a的值5传递给形参r，即r的值为5，那么a的值经过a=r+1后变成了6，而r的值经过r=r*2后变成了10，并返回，即在函数t中，变量x的值被赋值为10，那么在函数t中最后输出的时10+6=16。

采用引用调用时，由于形参r指向的是实参a的存储空间，即r与a指向的是同一块存储单元，首先a的值为5，经过a=r+1后变成了6，再经过r=r*2后变成了12，并返回，即在函数t中，变量x的值被赋值为12，那么在函数t中最后输出的时12+12=24。

试题4答案

（5）B（6）D

第 3 章：程序语言和语言处理程序基础知识　作者：希赛教育软考学院　来源：希赛网　2014年05月04日

考前冲刺

试题1

编译程序分析源程序的阶段依次是__(1)__。

（1）A．词法分析、语法分析、语义分析　B．语法分析、词法分析、语义分析

C．语义分析、语法分析、词法分析　D．语义分析、词法分析、语法分析

试题2

如图3-7所示的有限自动机中，0是初始状态，3是终止状态，该自动机可以识别__(2)__。

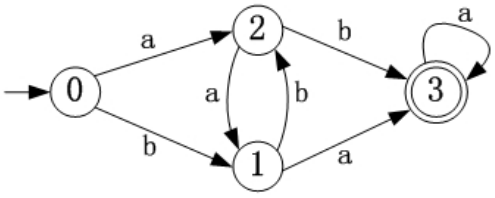


图3-7有限自动机示意图

（2）A．abab　B．aaaa　C．bbbb　D．abba

试题3

下图所示为两个有限自动机MI和M2 (A是初态、C是终态），__(3)__。

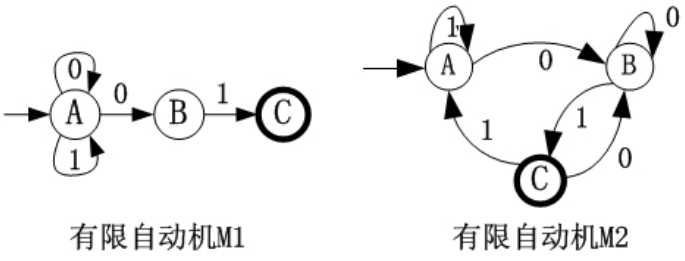


图3-8有限自动机示意图

（3）A．M1和M2都是确定的有限自动机

B．M1和M2都是不确定的有限自动机

C．M1是确定的有限自动机，M2是不确定的有限自动机

D．M1是不确定的有限自动机，M2是确定的有限自动机

试题4

以下关于可视化程序设计的叙述中，错误的是__(4)__。

（4）A．视化程序设计使开发应用程序无需编写程序代码

B．可视化程序设计基于面向对象的思想，引入了控件和事件驱动

C．在可视化程序设计中，构造应用程序界面就像搭积木

D．在可视化程序设计中，采用解释方式可随时查看程序的运行效果

试题5

以下关于汇编语言的叙述中，错误的是__(5)__。

（5）A．汇编语言源程序中的指令语句将被翻译成机器代码

B．汇编程序先将源程序中的伪指令翻译成机器代码，然后再翻译指令语句

C．汇编程序以汇编语言源程序为输入，以机器语言表示的目标程序为输出

D．汇编语言的指令语句必须具有操作码字段，可以没有操作数字段

试题6

如图3-9所示为一个有限自动机（其中，A是初态、C是终态），该自动机识别的语言可用正规式__(6)__表示。

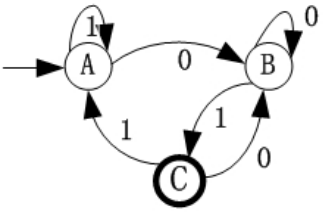


图3-9有限自动机示意图

（6）A．（0|1）*01　B．1*0*10*1　C．1*（0）*01　D．1*（0|10）*1*

试题7

算术表达式x-(y+c)*8的后缀式是__(7)__。（-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）

（7）A．x y c 8 - + *　B．x y - c + 8 *

C．x y c 8 * - +　D．x y c + 8 * -

试题8

传值与传地址是函数调用时常采用的信息传递方式，__(8)__。

（8）A．在传值方式下，是将形参的值传给实参

B．在传值方式下，形参可以是任意形式的表达式

考前冲刺

式

试题9

若一种程序设计语言规定其程序中的数据必须具有类型，则有利于__(9)__。

①在翻译程序的过程中为数据合理分配存储单元

②对参与表达式计算的数据对象进行检查

③定义和应用动态数据结构

④规定数据对象的取值范围及能够进行的运算

⑤对数据进行强制类型转换

（9）A．①②③　　B．①②④　　C．②④⑤　　D．③④⑤

试题10

下面C程序段中count++语句执行的次数为__(10)__。

```
for(int i=1；i<=11；i*=2)
    for(intj=1；j<=i；j++)
        count++；
```

（10）A．15　　B．16　　C．31　　D．32

试题11

算术表达式(a-b)*c+d的后缀式是__(11)__（-、+、*表示算术的减、加、乘运算，运算符的优先级和结合性遵循惯例）。

（11）A．a b c d - * + B．a b - c d * +

C．a b - c * d + D．a b c - d * +

上一节　　本书简介　　下一节

习题解析

试题1分析

编译程序分析源程序的阶段依次词法分析、语法分析、语义分析。详细情况请参看3.2.1。

试题1答案

（1）A

试题2分析

本题主要考查有限自动机。

在题目中，0是初始状态，3是终止状态，通过选项中的字符串可以从初始状态到达终止状态，则说明该字符串能被题目中的自动机识别。也可以理解为依次输入选项中的字符串，可以在该自动机中找到相应的路径。

对于选项A的字符串abab，通过ab可以达到终止状态，然后输入a任然可以有路径，但再输入b时，没有路径与其对应。因此A不可被该自动机识别。同样的道理，我们可以找到字符串aaaa能被该自动机识别。

试题2答案

（2）B

试题3分析

本题主要考查确定有限自动机与非确定有限自动机的判断。

非确定有限状态自动机与确定有限状态自动机的最大区别是它们的转移函数不同。确定有限状态自动机对每一个可能的输入只有一个状态的转移。非确定有限状态自动机对每一个可能的输入可以有多个状态转移，接受到输入时从这多个状态转移中非确定地选择一个。

在本题中给出的图M1中，我们可以看到当在状态A输入0时，它可以转移到它自己，也可以转移到状态B，所以M1是非确定的。而M2中不存在这样的情况，因此是确定的有限自动机。

试题3答案

（3）D

试题4分析

可视化程序设计主要是让程序设计人员利用软件本身所提供的各种控件，像搭积木式地构造应用程序的各种界面。可视化程序设计最大的优点是设计人员可以不用编写或只需编写很少的程序代码，就能完成应用程序的设计，这样就能极大地提高设计人员的工作效率。在可视化程序设计中，可随时查看程序的运行效果。

试题4答案

（4）A

试题5分析

面向机器的程序设计语言，使用汇编语言编写的程序，机器不能直接识别，要由一种程序将汇编语言翻译成机器语言，这种起翻译作用的程序叫汇编程序。汇编程序输入的是用汇编语言书写的源程序，输出的是用机器语言表示的目标程序。

试题5答案

（5）B

试题6分析

被有限自动机所识别是指从初态开始到终态结束，所输入的字符串能够按顺序地执行下去，若到某个状态不能往下走得到下一个字符，则认为不能识别。

在本题中，选项A能被识别。从初态A出发，不管经过多少个1和0之后，只能是处在A、B、C三种状态中的一种，所以在（0|1）*后，只能是处在A、B、C三种状态中的一种，不管是在那个状态，输入0后，都会处在状态B，然后输入1，都会转换到状态C，因此选项A能被该有限自动机所识别。

同样的道理，我们可以知道其它选项的正规式不能被识别。

试题6答案

（6）A

试题7分析

后缀表示也称为表达式的逆波兰表示。在这种表示方法中，将运算符号写在运算对象的后面，表达式中的运算符号按照计算次序书写。

对于表达式x-(y+c)*8，先计算y与c的和，再乘以8，最后用x减去这个这个计算，因此其后缀式为xyc+8*-。

试题7答案

（7）D

试题8分析

在函数调用时，系统为形参准备空间，并把实参的值赋值到形参空间中，在调用结束后，形参空间将被释放，而实参的值保持不变，这就是传值传递方式。传值传递方式中实参与形参之间的数

据传递是单向的，只能由实参传递给形参，因而即使形参的值在函数执行过程中发生了变化，也不会影响到实参值。在C语言中，当参数类型是非指针类型和非数组类型时，均采用传值方式。

传地址方式把实参的地址赋值给形参，这样形参就可以根据地址值访问和更改实参的内容，从而实现双向传递。当参数类型是指针类型或数组类型时，均采用传地址方式。

区别于参数传值方式和返回值传递方式，传地址方式具有以下明显的优势。

（1）参数传值方式是主调函数与被调函数之间的单向数据传递方式，而参数的传地址方式则实现了二者之间的双向数据传递。

（2）函数的返回值每次只能把一个数据项从被调函数传递到主调函数，而参数的传地址方式却可一次性地传递多个数据项到主调函数。

试题8答案

（8）C

试题9分析

一种程序设计语言规定其程序中的数据必须具有类型，好处如下：

（1）有利于在翻译程序的过程中为数据合理分配存储单元，因为程序设计语言为不同的数据类型规定了其所占的存储空间，如果数据类型确定，其所占的存储空间也是确定的。

（2）有利于对参与表达式计算的数据对象进行检查，因为知道数据的数据类型，我们就可以根据类型来判断该数据是否可以参与某表达式计算，如自加、自减的操作数不允许是浮点数，这只要根据数据的类型就能判断某操作数，是否能进行自加、自减运算。

（3）有利于规定数据对象的取值范围及能够进行的运算，根据数据类型，我们可以数据的存储空间，也同时能知道数据的表示范围，如C语言中的整型数据，它占两个字节（16位），能表示的数据范围就是-216至216-1。

综上所述，可知本题的正确答案选B。

试题9答案

（9）B

试题10分析

本题中给出的是一个双重循环结构，循环体就是count++。第一层循环的循环次数为4次，分别为i=1,2,4,8的情况。而当i=1时，第二层循环循环1次；当i=2时，第二层循环2次；当i=4时，第二层循环4次；当i=8时，第二层循环8次。那么可知循环体一共执行了1+2+4+8=15次。

试题10答案

（10）A

试题11分析

本题要求通过中缀表达式，求后缀式（也称为逆波兰）。解答这类问题，可以借助于二叉树（第4章数据结构中有关于二叉树的详细内容）。因为中缀表达式对应于一颗二叉树的中序遍历，前缀表达式对应于二叉树的前序遍历，后缀表达式对应于二叉树的后序遍历。所以在本题中，需要先把二叉树构造出来。将表达式(a-b)*c+d构造成二叉树，如图所示。

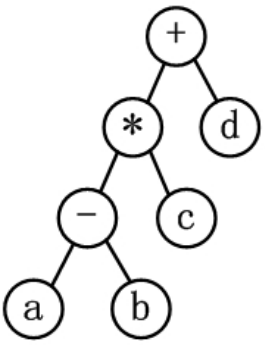


图3-10二叉树示意图

将此树进行后序遍历，得到：ab-c*d+。

试题11答案

（11）C

上一节　本书简介　下一节

考点脉络

数据结构是指数据对象及其相互关系和构造方法。在软件设计过程中，不同的数据结构的选用，对系统最终效果的影响极大。所以该知识点是软件设计师核心考点，无论是上午综合知识部分，还是下午的软件设计部分，考查分值都很高。

根据考试大纲，本章要求考生掌握数组、链表、队列和栈、树、图、杂凑相关知识，从历年的考试情况来看，本章主要考查常见数据结构的逻辑结构特性及存储的相关内容。

上一节　本书简介　下一节

数组与线性表

按数据的逻辑结构来划分，常见的数据结构包括：数组（静态数组、动态数组）、线性表（顺序表、链表、队列、栈）、树（二叉树、查找树、平衡树、线索树、堆）、图。本节将介绍数组与线性表的相关内容。

上一节　本书简介　下一节