

一手资源 持续更新 认准淘宝旺旺ID：蔚然科技学堂 或者：君学赢精品课堂  
如在其他店购买请差评或退款，他们断更新且残缺。可找我店免费领完整新资料

## 2 程序语言基础知识



# 本节主要考点

- 1、编译程序和解释程序
- 2、程序语言的数据成分和控制成分
- 3、编译程序的过程
- 4、中缀、前缀与后缀表达式



# 程序语言概述

## 1、低级语言和高级语言：

- 低级语言：机器语言和汇编语言。是一种面向机器的语言，其格式取决于计算机的机器指令。难以理解，程序可读性差，程序设计效率低。
- 高级语言：面向各类应用的程序语言。如Java、C、C++、Python、PHP、JavaScript等等。与人们使用的语言较为接近，便于理解，提高了程序设计的效率。

## 2、汇编、解释、编译：

高级程序语言必须进行翻译才能为计算机硬件所理解，常用的翻译方式有汇编、解释和编译。

- 用汇编语言编写的：需要汇编程序翻译成目标程序，然后执行目标程序。
- 用高级语言编写的：需要解释程序或编译程序进行翻译，然后再运行。



# 程序语言概述

## 3、编译程序和解释程序：

(1) 解释程序（解释器）：要么直接解释执行源程序，要么将源程序翻译成某种中间代码后再加以执行。它按源程序中语句的执行顺序，逐条翻译并立即执行相关功能。

(2) 编译程序（编译器）：将源程序翻译成目标程序（目标代码），然后再在计算机上运行目标程序。一般分为两个阶段：

编译阶段：把源程序翻译成目标程序。

运行阶段：执行目标程序。



# 程序语言概述

## 4、编译和解释的区别：

- 编译方式下，机器上运行的是与源程序等价的目標程序，源程序和编译程序都不再参与目标程序的执行过程。
- 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。

简单来说，在解释方式下，翻译源程序时不生成独立的目標程序，而编译器则将源程序翻译成独立保存的目標程序。

## 5、编译和解释的比较：

- 编译比解释方式可能取得更高的效率。
- 解释方式比编译方式更灵活
- 解释方式可移植性好。



# 程序语言的数据成分

- 1) **常量和变量**：按照程序运行时数据的值能否改变，将程序中的数据分为常量和变量。
- 2) **全局变量和局部变量**：按数据的作用域范围，可将其分为全局变量和局部变量。系统为全局变量分配的存储空间在程序运行的过程中一般是不改变的，而为局部变量分配的存储单元是可以动态改变的。
- 3) **数据类型**：
  - 基本类型：整型、字符型、实型和布尔类型。
  - 特殊类型：空类型。
  - 用户定义类型：枚举类型。
  - 构造类型：数组、结构、联合。
  - 指针类型：type \*。
  - 抽象数据类型：类类型。



# 程序语言的控制成分

## 1) 顺序结构

- 计算过程从所描述的第一个操作开始，按顺序依次执行后续的操作，直到序列的最后一个操作。

## 2) 选择结构

- 选择结构提供了在两种或多种分支中选择其中一个的逻辑。

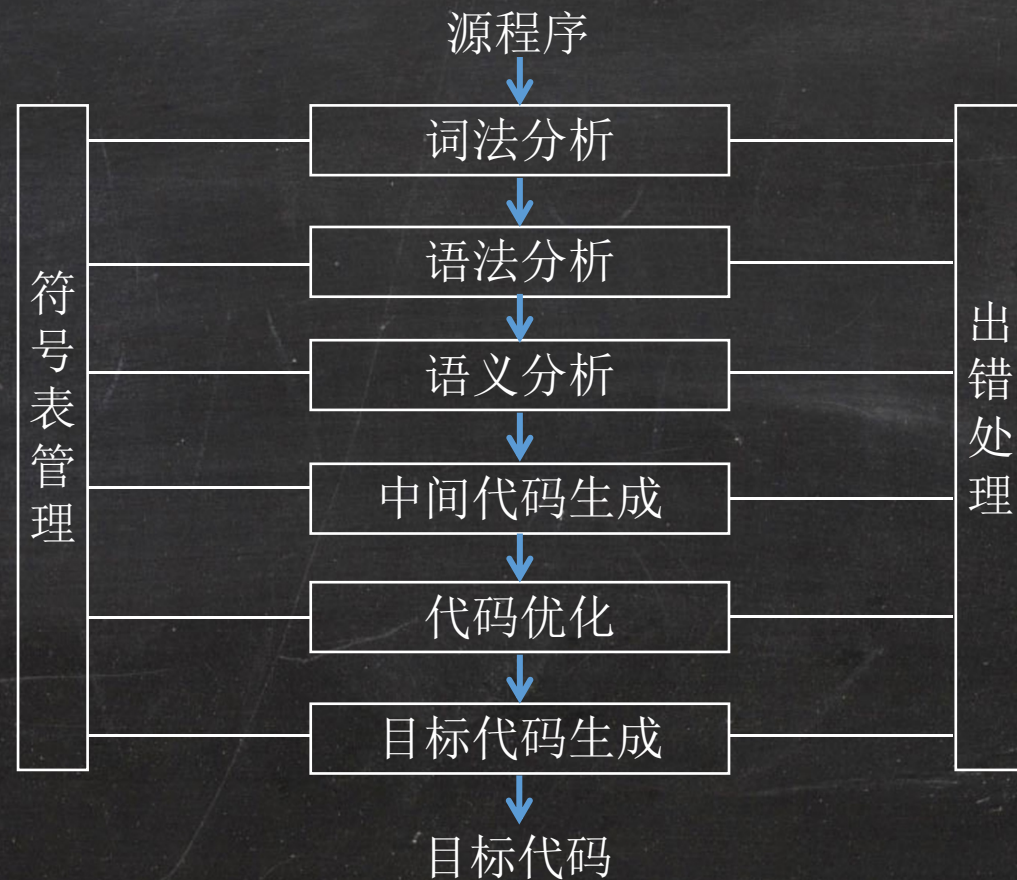
## 3) 循环结构

- 循环结构描述了重复计算的过程，通常由三部分组成：初始化、循环体和循环条件。



# 编译过程

- 编译程序的作用是把某种高级语言书写的源程序翻译成与之等价的目标程序，其工作过程一般可以分为6个阶段：





# 编译过程

## 1、词法分析：

- 任务：对源程序从前到后（从左到右）逐个字符地扫描，从中识别出一个个“单词”符号。“单词”符号是程序设计语言的基本语法单位，如关键字（保留字）、标识符、常数、运算符和标点符号、左右括号等。

## 2、语法分析：

- 任务：根据语言的语法规则，在词法分析的基础上，分析单词串是否构成短语和句子，即是否为合法的表达式、语句和程序等基本语言结构，同时检查和处理程序中的语法错误。
- 如果源程序没有语法错误，语法分析后就能正确地构造出其语法树。



# 编译过程

## 3、语义分析

- 任务：分析各语法结构的含义，检查源程序是否包含静态语义错误，并收集类型信息供后面的代码生成阶段使用。
- 语义分析的一个主要工作是进行类型分析和检查。一个数据类型一般包含两个方面的内容：类型的载体及其上的运算。例如：整除取余运算符只能对整型数据进行运算，如果运算对象为浮点数，则认为是一种类型不匹配的错误。

## 4、中间代码生成

- 任务：根据语义分析的输出生成中间代码。
- 常用的中间代码有：后缀式（逆波兰式）、四元式（三地址码）、树形表示。



# 编译过程

## 5、代码优化：

- 由于编译器将源程序翻译成中间代码的工作是机械的、按固定模式进行的，因此，生成的中间代码往往在时间和空间方面的效率较差。当需要生成高效的目标代码时，就必须进行优化。
- 优化过程可以在中间代码生成阶段进行，也可以在目标代码生成阶段进行。

## 6、目标代码生成

- 目标代码生成是编译器工作的最后一个阶段。
- 任务：把中间代码变换成特定机器上的绝对指令代码、可重定位的指令代码和汇编指令代码。
- 这个阶段的工作与具体的机器密切相关。



# 编译过程

## 7、符号表管理

- 符号表的作用是记录源程序中各种符号的必要信息，以辅助语义的正确性检查和代码生成，在编译过程中需要对符号表进行快速有效地查找、插入、修改和删除等操作。

## 8、出错处理

源程序中不可避免地会有一些错误，大致分为静态错误和动态错误。

- (1) 动态错误发生在程序运行时，如：变量取零时作除数、引用数组下标错误等。
- (2) 静态错误是指编译阶段发现的程序错误，可分为语法错误和静态语义错误。
  - ① 语法错误：单词拼写错误、标点符号错、表达式中缺少操作数、括号不匹配等有关语言结构上的错误。
  - ② 静态语义错误：语义分析时发现的运算符和运算对象类型不合法等错误。



# 中缀、前缀与后缀表达式

- 中缀表达式：即我们通常所使用的表达式。如  $(a+b)*c-d$
- 前缀表达式（波兰式）：将运算符写在前面，操作数写在后面，且不使用括号。  $-*+abcd$
- 后缀表达式（逆波兰式）：将运算符写在后面，操作数写在前面，且不使用括号。  $ab+c*d-$



# 中缀、前缀与后缀表达式

- 三种表达式之间的转换：

例：  $(a+b)*c-d$

(1) 中缀表达式转前缀表达式：

- ① 按计算顺序全部加上括号：  $((a+b)*c)-d$
- ② 把每一对括号内的运算符移到括号前面：  $-(* (+ (ab) c) d)$
- ③ 把所有括号去掉：  $-*+abcd$

(2) 中缀表达式转后缀表达式：

- ① 按计算顺序全部加上括号：  $((a+b)*c)-d$
- ② 把每一对括号内的运算符移到括号后面：  $(( (ab) +c) *d) -$
- ③ 把所有括号去掉：  $ab+c*d-$



# 传值调用和引用调用

- 函数调用时基本的参数传递方式有两种：传值调用和引用调用。

## (1) 传值调用：

- 信息传递是单向的，只能将实参的值传递给形参，而形参不能再将值传递给实参。
- 实参可以是常量（表达式），也可以是变量（数组元素）。

例：

```
int sum(int x, int y) {  
    int z;  
    z=x+y;  
    return z; }  
}
```

函数调用时：sum(2, 3);



# 传值调用和引用调用

## (2) 引用调用（传地址调用）：

- 在这种方式下，可以认为形参名实际上是实参的别名，被调函数中对形参的访问和修改实际上就是对实参的访问和修改。因此客观上可以实现双向传递。
- 因此只能是变量（数组元素），而不能是常量（表达式）。

例： `void swap(int &x, int &y) {`  
    `int temp;`  
    `temp=x; x=y; y=temp }`

函数调用时： `sum(a, b);`



## 2 程序语言基础知识

【12年第22题】算术表达式 $x-(y+c)*8$ 的后缀式是（ ）（-、+、\*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）。

- A.  $x\ y\ c\ 8\ -\ +\ *$       B.  $x\ y\ -\ c\ +\ 8\ *$       C.  $x\ y\ c\ 8\ *\ +\ -$       D.  $x\ y\ c\ +\ 8\ *\ -$

【13年第20题】以下关于解释程序和编译程序的叙述中，正确的是（ ）。

- A. 编译程序和解释程序都生成源程序的目标程序
- B. 编译程序和解释程序都不生成源程序的目标程序
- C. 编译程序生成源程序的目标程序，解释程序则不然
- D. 编译程序不生成源程序的目标程序，而解释程序反之

【13年第21题】以下关于传值调用与引用调用的叙述中，正确的是（ ）。

- ① 在传值调用方式下，可以实现形参和实参间双向传递数据的效果
  - ② 在传值调用方式下，实参可以是变量，也可以是常量和表达式
  - ③ 在引用调用方式下，可以实现形参和实参间双向传递数据的效果
  - ④ 在引用调用方式下，实参可以是变量，也可以是常量和表达式
- A. ①③      B. ①④      C. ②③      D. ②④

【13年第22题】在对高级语言源程序进行编译的过程中，为源程序中变量所分配的存储单元的地址属于（ ）

- A. 逻辑地址      B. 物理地址      C. 接口地址      D. 线性地址



## 2 程序语言基础知识

【14年第20题】以下程序设计语言中，（ ）更适合用来进行动态网页处理。

- A. HTML                      B. LISP                      C. PHP                      D. JAVA/C++

【14年第21题】引用调用方式下进行函数调用，是将（ ）。

- A. 实参的值传递给形参                      B. 实参的地址传递给形参  
C. 形参的值传递给实参                      D. 形参的地址传递给实参

【14年第22题】编译程序对高级语言源程序进行编译的过程中，要不断收集、记录和使用源程序中一些相关符号的类型和特征等信息，并将其存入（ ）中。

- A. 符号表                      B. 哈希表                      C. 动态查找表                      D. 栈和队列

【15年第20题】以下关于程序设计语言的叙述中，错误的是（ ）。

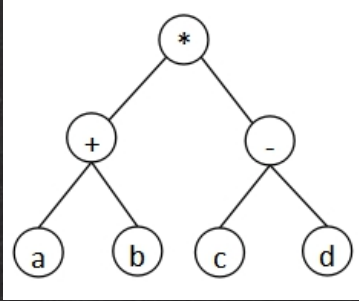
- A. 程序设计语言的基本成分包括数据、运算、控制和传输等  
B. 高级程序设计语言不依赖于具体的机器硬件  
C. 程序中局部变量的值在运行时不能改变  
D. 程序中常量的值在运行时不能改变



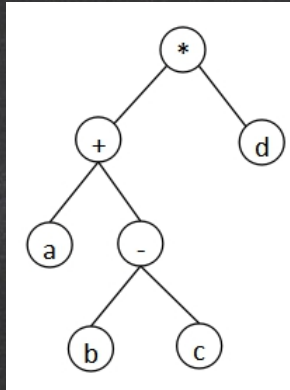
## 2 程序语言基础知识

【15年第21题】与算术表达式“(a+ (b-c)) \*d”对应的树是 ( )。

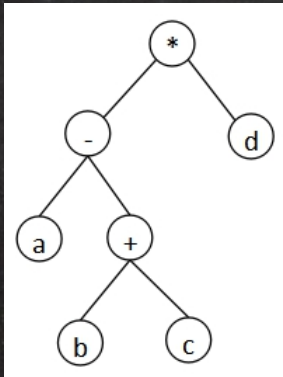
A.



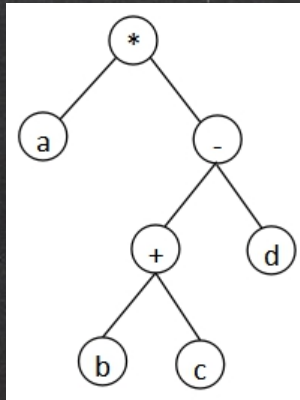
B.



C.



D.



【15年第22题】程序中全局变量的存储空间在 ( ) 分配。

A. 代码区

B. 静态数据区

C. 栈区

D. 堆区



## 2 程序语言基础知识

【16年第20题】以下关于高级程序设计语言实现的编译和解释方式的叙述中，正确的是（ ）。

- A. 编译程序不参与用户程序的运行控制，而解释程序则参与
- B. 编译程序可以用高级语言编写，而解释程序只能用汇编语言编写
- C. 编译方式处理源程序时不进行优化，而解释方式则进行优化
- D. 编译方式不生成源程序的目标程序，而解释方式则生成

【16年第21题】以下关于脚本语言的叙述中，正确的是（ ）。

- A. 脚本语言是通用的程序设计语言
- B. 脚本语言更适合应用在系统级程序开发中
- C. 脚本语言主要采用解释方式实现
- D. 脚本语言中不能定义函数和调用函数

【16年第22题】将高级语言源程序先转化为一种中间代码是现代编译器的常见处理方式。常用的中间代码有后缀式、（ ）、树等。

- A. 前缀码
- B. 三地址码
- C. 符号表
- D. 补码和移码



## 2 程序语言基础知识

【17年第20题】在高级语言源程序中，常需要用户定义的标识符为程序中的对象命名，常见的命名对象有（ ）。

- ①关键字（或保留字） ②变量 ③函数 ④数据类型 ⑤注释  
A. ①②③ B. ②③④ C. ①③⑤ D. ②④⑤

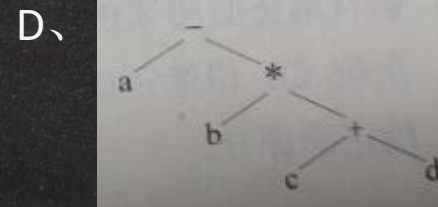
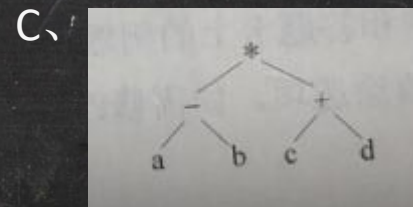
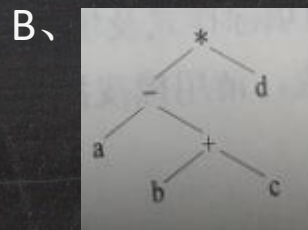
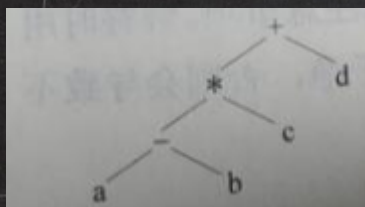
【17年第22题】在以阶段划分的编译中，判断程序语句的形式是否正确属于（ ）阶段的工作。  
A. 词法分析 B. 语法分析 C. 语义分析 D. 代码生成

【18年第5、6、7题】算术表达式采用后缀式表示时不需要使用括号，使用（ ）就可以方便地进行求值。  
 $a - b * (c + d)$ （其中， $-$ 、 $+$ 、 $*$ 表示二元算术运算减、加、乘）的后缀式为（ ），与该表达式等价的语法树为（ ）。

(5) A. 队列 B. 数组 C. 栈 D. 广义表

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

(7) A、





## 2 程序语言基础知识

【19年第22题】以下关于编译和解释的叙述中，正确的为（ ）。

- ① 编译是将高级语言源代码转换成目标代码的过程
  - ② 解释是将高级语言源代码转换为目标代码的过程
  - ③ 在编译方式下，用户程序运行的速度更快
  - ④ 在解释方式下，用户程序运行的速度更快
- A. ①③      B. ①④      C. ②③      D. ②④

【19年第23题】函数调用和返回控制是用（ ）实现的。

- A. 哈希表    B. 符号表    C. 栈    D. 优先队列

【19年第24题】通用的高级设计语言一般都会提供描述数据、运算、控制和数据传输的语言成分，其中，控制包括顺序、（ ）和循环结构。

- A. 选择    B. 递归    C. 递推    D. 函数

【20年第22~24题】程序设计语言的基本成分包括数据、运算、控制和（ ）。数据是程序操作的对象，按照数据组织形式可以分为多种类型，其中枚举属于（ ）类型；数组属于（ ）类型。

- (22) A. 存储    B. 分配    C. 传输    D. 函数  
(23) A. 基本    B. 用户定义    C. 构造    D. 特殊  
(24) A. 基本    B. 用户定义    C. 构造    D. 特殊



## 2 程序语言基础知识

【21年第22~23题】在程序设计语言中，（ ）表示了构成语言的各个记号和使用者的关系，而语境是指理解和实现程序设计语言的环境，包括（ ）环境和运行环境。

(22) A.语法 B.语义 C.语用 D.词法

(23) A.开发 B.调试 C.测试 D.编译

【21年第24题】在程序编译过程中，执行类型分析和检查是在（ ）阶段。

A.词法分析 B.语法分析 C.语义分析 D.代码优化



## 2 程序语言基础知识

### 【22年第6题】

计算机在处理算术表达式 $78+21*(36-34)$ 时，先将其转换成“（ ）”的后缀形式表示，然后利用（ ）进行计算。

(5) A.  $78\ 21+36*34-$       B.  $78\ 21\ 36\ 34-*+$       C.  $78\ 21\ 36\ 34+*-$       D.  $36\ 34-21*78+$

(6) A. 栈      B. 队列      C. 数组      D. 串

### 【22年第22题】

系统为（ ）分配的存储空间在程序运行过程中一般是不改变的。  
A. 结构体      B. 数组      C. 全局变量      D. 局部变量

### 【22年第23题】

以下关于C语言函数的说法中，不正确的是（ ）。

- A. 函数是一段具有独立功能的程序单元
- B. 函数是先声明后引用
- C. 函数的定义包括函数首部和函数体
- D. 函数允许嵌套定义

### 【22年第24题】

（ ）是一种解释性、面向对象、动态类型的脚本语言。  
A. C      B. C++      C. Ruby      D. JavaScript