高级语言开发的程序不能直接运行

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

**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**

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，然后在运行

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

1. 语法分析要在词法分析的基础上，根据语法规则将单词符号分解成各类语法单位
2. 语法分析能发现语法错误及其结构是否合法，但不能发现语义错误
3. 语义分析阶段主要检查源程序是否包含静态语意错误，并收集类型信息供后面代码用
4. 目标代码生成是把中间代码变成特定机器上的绝对指令代码等，与具体机器机器有关

错题：

**在C程序中，若在某个表达式中引用了未赋初值的变量（全局变量和静态局部变量自动初始化为0，普通局部变量是一个随机数），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不一定会报异常**

**（全局变量经过方法改变了值会保存）数据流起点必须是加工 T**

**不管是常理 还是变量，它们都有其类型属性**

**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**

**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**

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，叫汇编程序

1. 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
2. 汇编语言语句分为指令语句（产生相应机器码）、伪指令（指示汇编程序在汇编源程序时完成某操作，不产生机器代码）、宏指令

高级语言源程序经过编译->可执行文件，但不能把可执行文件编程高级语言源代码，只能转换成汇编程序

交叉编译：在一个平台运行的程序能译出在另一个平台可以运行的代码

数据类型

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

目的

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

表达式

（1）中缀表达式

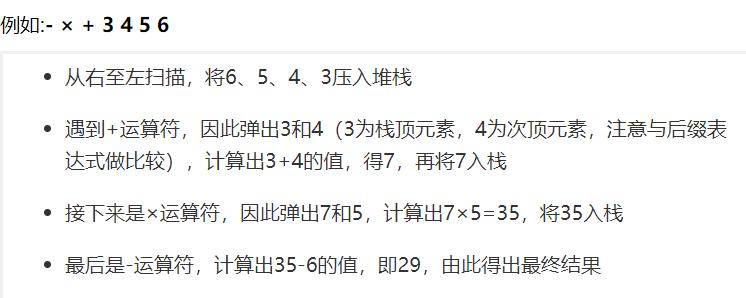
由于中缀表示法仅适合于二元操作符，一种语言不能只使用中缀表示法，而必须结合中缀与前 缀表示法。这种混合使用会使翻译过程相对复杂

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

中缀表达式就是常见的运算表达式，如(3+4)×5-6

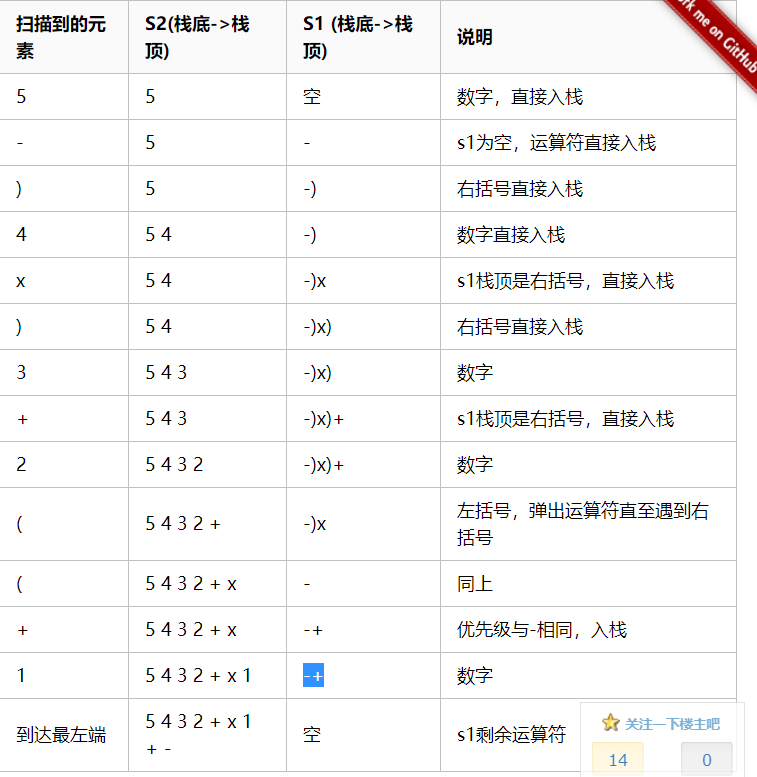
（2）前缀表达式（波兰式）**算符位于操作数之前**

公式(a+b)\*(a-b)，使用前缀表达式表示将变为 \* + a b – a b



栈顶元素 运算符 次顶元素

**1+((2+3)×4)-5转为前缀表达式**



（3）后缀表达式（逆波兰式）

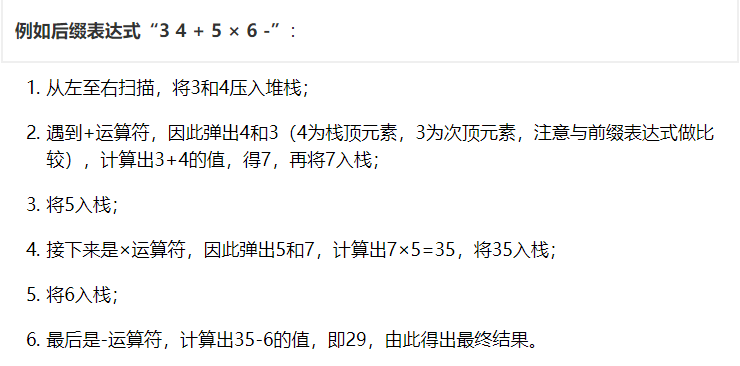
规则：

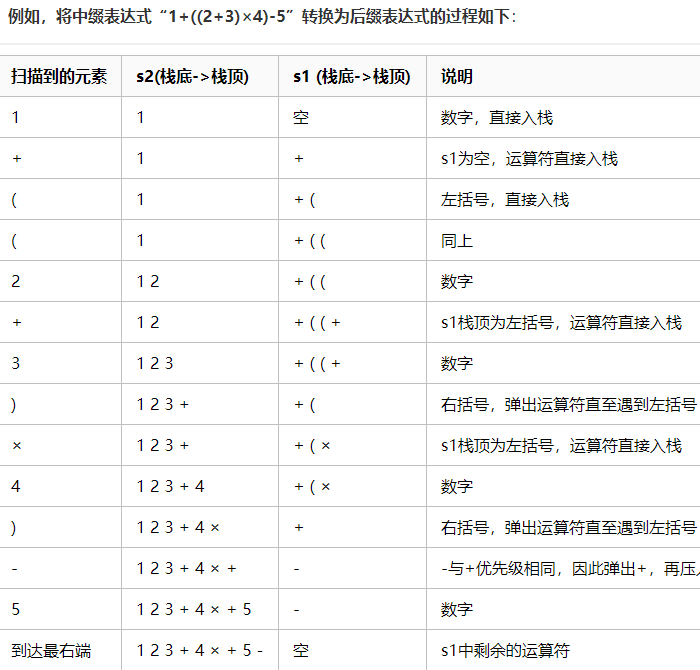
旧>新 则弹出 旧<新则不变

逻辑与>或

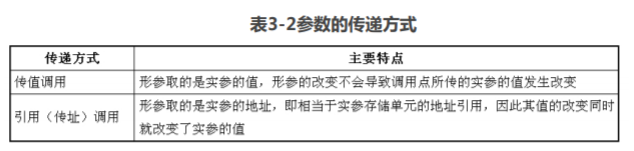
操作符跟在操作数之后

a b + a b– \*





程序调用的实现机制



传值：实参额形参是两个独立实体，实参可以是数组

传址：形参和实参占用相同的存储单元，数据双向传递，但实参不能是常量或表达式（相当于传值）

程序三种基本控制结构：顺序、选择、重复

错误分为静态和动态错误：

静态：编译阶段发现，分为语法错误（单词拼写）和静态语义错误（运算符与运算对象不合法）

动态：程序运行发现，除数为0，数组元素下标错误

**语言分类**

面对象：

Python：编写独立程序，快速脚本

Java：开发internet英语，具有软硬件独立性和交互能力

C#：与com（组建对象模型）直接集成

可视化程序设计：引入了控件和事件驱动(T)

1.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

3.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

3.在可视化程序设计中，可随时查看程序的运行效果。

逻辑型：

Prolog：一系列事实关系规则的集合，适用于证明等问题程序

逻辑运算符(无括号时左开始还是又开始计算)

左结合： a+b+c ->(a+b)+c

右结合：x=y=z -> y=z再x=y

脚本语言（动态语言）与编程语言区别：  
1.（脚本）语言严格复杂些，有脚本引擎执行（需解释器即是解释性语言），即不能翻译成二进制代码以可执行文件形式存在

2.函数两者类似

**算法**

自动机：

可识别：能从初态走到终态

确定：每一个输入只有一个状态转移

非确定： 多个~

正规表达式： a的\*次方 \*为0个或多个

上下文无关：S非终结符 s终结符