

四、软件与环境

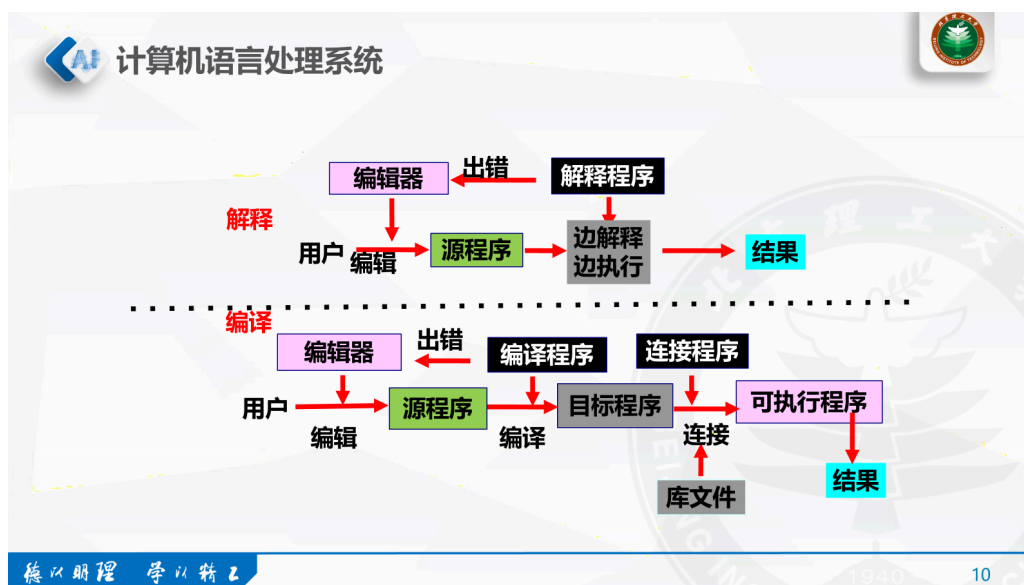
1. 程序设计

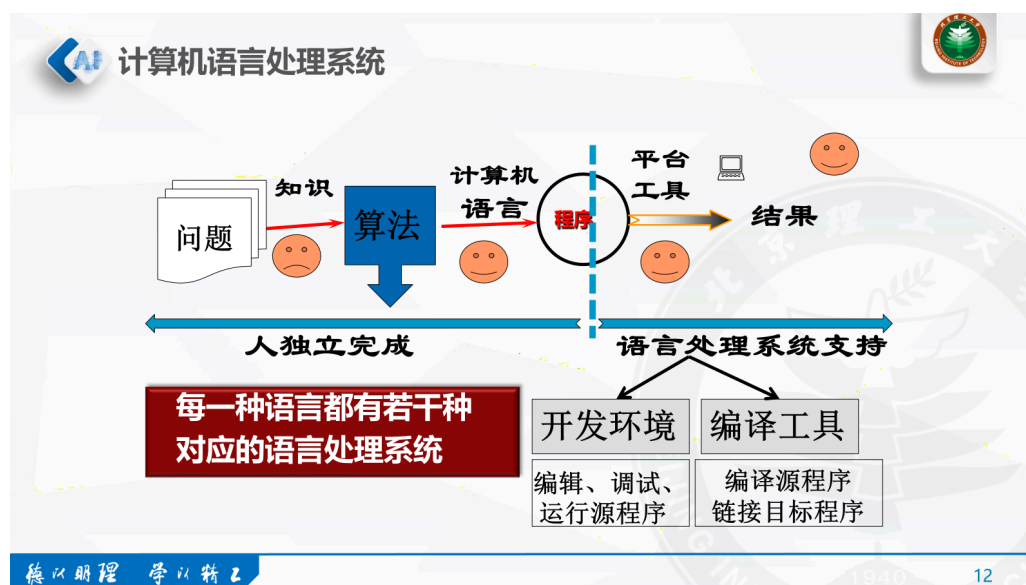
1.1 面向过程程序设计特点

结构化程序设计方法是面向过程编程应遵循的基本方法和原则。主要包括：

1. 只采用三种基本的程序控制结构（顺序、选择、循环）来编制程序，从而使程序具有良好的结构；
2. 程序设计自顶而下；
3. 用结构化程序设计流程图表示算法。

1.2 计算机语言处理系统





2. 算法基础

2.1 图灵机与可计算

2.1.1 图灵机

图灵机包括以下四个部分：

1. 一条**无限长的纸带**，用于使用二进制符号来表达计算所用的数据和控制规则；
2. 一个**读写头**，用于获取或者改写纸带当前位置上的符号；
3. 一个**状态寄存器**，用于保存图灵机当前所处的状态（包括停机状态）；
4. 一套**控制规则**，它根据当前机器所处的状态以及当前读写头所获取的符号，来确
定读写头下一步的动作，并改变状态寄存器的值，令机器进入一个新的状态。

2.1.2 可计算

可计算性是指对于给定的问题，是否存在一种算法可以解决它。

不可计算问题与可计算判断

什么问题不可计算

图灵论题

一个可计算问题是“当且仅当它在图灵机上经过有限步骤之后可以得到正确的结果”

根据这一论题，通常人们把所面临的问题分为可计算问题与不可计算问题两大类

德以明理 学以致用 21

不可计算问题与可计算判断

问题的可计算判断

“能行过程”：针对所要解决的问题是否存在能行方法，以此来判断可计算问题是否实际可解。

无论是数学还是工程，解决问题的过程就是问题状态发生变化的过程。如果我们以参数形式来描述问题状态，那么解决问题的过程就可以看作是一个参数变化的过程，如表所示。这个过程中如果输入参数和输出参数的对应关系是明确的，则说明这个过程是能行的，也就是说这个问题是可计算的。


过程时间	问题状态	参数形式
开始	初始状态	输入参数
结束	结果状态	输出结果

德以明理 学以致用 24

2.2 算法复杂度

2.2.1 时间复杂度

计算所需的步数或指令条数.



计算复杂性

算法的时间度量
算法的所需时间与问题规模的函数 $T(n)$

(a) $X=X+1$
(b) for ($i=1$; $i \leq n$; $i++$)
 $X=X+1$
(c) for ($i=1$; $i \leq n$; $i++$)
 for ($j=1$; $j \leq n$; $j++$)
 $X=X+1$

以上操作执行的次数分别是
 $1, n, n^2$

德以明理 学以精工 1940 29

2.2.2 空间复杂度


计算所需的存储空间大小。

2.3 算法的特点


算法具有以下基本特征：

- 有穷性：一个算法必须在执行有限个操作步骤后终止
- 确定性：算法中每一步的含义必须是确切的，不可出现任何二义性
- 有效性：算法中的每一步操作都应该能有效执行，一个不可执行的操作是无效的
- 有零个或多个输入：这里的输入是指在算法开始之前所需要的初始数据，输入的多少取决于特定的问题
- 有一个或多个输出：所谓输出是指与输入有某种特定关系的量，在一个完整的算法中至少会有一个输出。








2.4 流程图表示算法



算法的表示方法




标准流程图符号含义


符号名称	符 号	功 能
起止框		表示算法的开始和结束
输入/输出框		表示算法的输入/输出操作，框内填写需输入或输出的各项
处理框		表示算法中的各种处理操作，框内填写处理说明或算式
判断框		表示算法中的条件判断操作，框内填写判断条件
注释框		表示算法中某操作的说明信息，框内填写文字说明
流程线		表示算法的执行方向
连接点		

德以明理 学以精工

194038



算法的表示方法

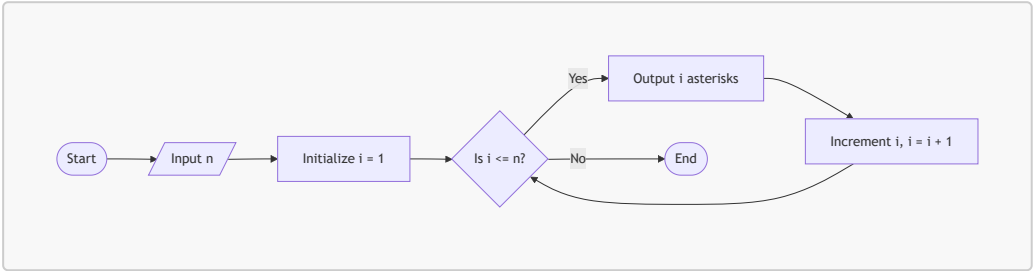


练习：
用传统流程图表示如下问题的算法
从键盘输入n值，输出n行用”*”号组成的直角三角形。
输入：n
输出：三角形

示例3：
输入：5
*
**

德以明理 学以精工

194042





典型问题的算法设计



成绩排名问题—排序算法

问题描述： 在一个班级有30名同学，一次考试每个同学有一个考试成绩，如何将这30名同学的成绩由低至高进行排序？

问题分析： 这是一个排序问题。一般认为，日常的数据处理中有1/4的时间应用在排序上，据不完全统计，到目前为止的排序算法有上千种。

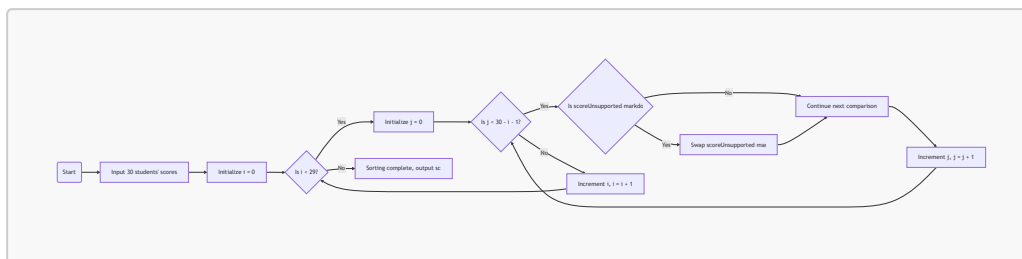
算法设计：

- 1、用冒泡排序解决方案
- 2、用选择排序解决方案
- 2、用插入排序解决方案

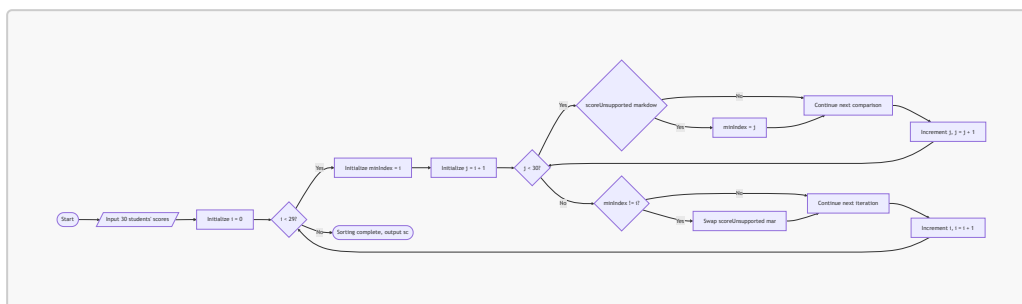
德以明理 学以致用

45

冒泡排序



选择排序



插入排序

