



数据结构

什么是数据结构&数据结构的应用举例



什么是数据结构 ??

- 对于什么是数据结构，我觉得应该分成两部分来理解，也就是说数据结构包括了数据、结构两部分的内容。

什么是数据？？

- 是描述客观事物的符号，是计算机中可以操作的对象，是能被计算机识别，并输入给计算机处理的符号集合。这个官方定义不是很容易理解，用我们自己的话说，数据就是计算机能处理的东西，比如说整型、实数型，当然，还包括字符、声音、图像...比如说我们常用的搜索引擎，一般会有网页、MP3、图片、视频等分类。其中，MP3是声音数据，图片是图像数据。说白了，我们所说的数据就是符号，但其必须具有两个属性：
 - 1、可以输入到计算机中
 - 2、能被计算机处理

什么是结构？？

- 结构，简单来说就是数据之间的关系。我们都知道，在现实中，不同数据之间不是独立的，而是存在某种特定的关系，我们称这些关系为结构。那么在计算机中也是如此，计算机中的数据并不是孤立的、杂乱无章的，而是具有内在联系的数据集合。我们要写一个好的程序，就要分析处理对象的特性及各处理对象之间存在的关系。我们知道了结构就是关系，那么都有那些具体的关系呢？
- 按照视点的不同，我们把数据结构分为逻辑结构和物理结构。

逻辑结构

- 逻辑结构：是指数据对象中数据元素之间的相互关系。
- 集合关系：集合结构中的数据元素除了同属于一个集合外，它们之间没有其他关系。就是类似与我们数学中所说的集合，各元素之间没有关系，彼此平等。
- 线性结构：线性结构中的数据元素之间是一一对应的关系。就像排队一样，我们只和我们前面的人有关系，和其他人没有关系。
- 树形结构：树形结构中的元素之间存在一种一对多的层次关系。就像我们所见的数一样，从下往上，枝叶逐渐增多，但下一层只和上一层有关系。
- 图形结构：图形结构的数据元素是多对多的关系，就像地图一样，不同的节点之间都会有关系。
- 逻辑结构是针对具体问题的，是为了解决某个问题，在对问题理解的基础上，选择一个合适的数据结构表示数据之间的逻辑关系，我们要懂得根据实际的问题来选择合适的结构来组织我们的数据。

物理结构

物理结构：是指数据的逻辑结构在计算机中的存储形式。说白了，选定了具体的逻辑结构后，我们要怎么样把数据存在计算机中。数据的物理结构应该正确反映数据元素之间的逻辑关系。如何在计算机中存储元素之间的逻辑关系，是实现物理结构的重点和难点。

顺序存储结构：把数据元素存放在地址连续的存储单元中。这种存储结构很简单，就类似与我们开的数组，大家按顺序排好队，每个人占一端空间，大家都不插队。这种方式存储数据，最大的好处是易于遍历，大家都在一起，当然容易遍历啦，但是删除操作就很麻烦，需要把要删除的数字后面的数都往上移位。

链式存储结构：是把数据元素存放在任意的存储单元里，这组存储单元不一定要连续的。哪要怎么从找到下一个位置呢？这时候就需要指针上场了。这种存储方式比较灵活，数据存放在那不重要，只要一个指针存放了相应的地址就能找到。但查找就很困难了，因为每一次查找都必须从头开始查找。

如何学习

- 通过上面的描述，我们知道了要学习的内容，那么学习的重点难点也就出来了，那就是怎么通过对具体问题的分析，从而采用合适的元素及数据的逻辑结构，再者，我们又怎么通过确定的逻辑结构来进一步选择合适的物理结构。总的一句话就是说，如何让逻辑结构体现问题的结构，如何让物理结构体现逻辑结构。
- 我们已经知道了重点难点，那最好的学习方法就是逐个击破啦：
 - 平时要多思考问题，思考问题需要的操作都有那些，选择什么样的逻辑结构更合适
 - 在选择物理结构时，也要思考要怎样才能更好的反映逻辑结构
 - 要多敲代码实现，从实践中发现问题

应用举例

表达式求值

表达式求值是高级语言编译中的一个基本问题，是栈的典型应用实例。

$$1+2*3/(1+5)=2$$

任何一个表达式都是由数（操作数）和符号（运算符 + - * /，界限符（））组成。
而符号又有优先级，优先计算得到的值再作为数继续运算，当符号用尽后，值即为所得。

$$1+2*3/(1+5) \rightarrow 1+6/(1+5) \rightarrow 1+6/6 \rightarrow 1+1 \rightarrow 2$$

优先级规则：

（1）先左后右

（2）先乘除，后加减

（3）先括号内，后括号外

此时需要两个栈，一个称作operator，用以存放符号。一个称作operand，用于存放操作数或中间结果。

算法实现

算法的基本过程如下：

初始化两个栈，然后讲表达式起始符 “#” 压入operator栈。

输入表达式，以 “#” 结尾。

依次读取表达式，读取过程中，若是操作数，直接入栈operand。若是符号，则与operator栈顶运算符进行比较。

（1）若 栈顶符<刚读运算符：刚读运算符 进栈；

（2）若 栈顶符>刚读运算符：进行计算，operator栈顶符退栈，送入x，同时将operand栈退栈两次，得到a，b，对a，b，x进行运算后，将结果作为中间结果推入operand栈；

（3）若 栈顶符=刚读运算符：说明是左右括号相遇，栈顶符退栈即可；

当operator栈的栈顶符和当前符都为 “#” 时，说明求值完毕。