阅读：32,232　　作者：解学武

# 什么是栈，栈及其特点和应用详解

上一节　　下一节

同顺序表和链表一样，栈也是用来存储逻辑关系为 "一对一" 数据的线性存储结构，如图 1 所示。


图 1 栈存储结构示意图

从图 1 我们看到，栈存储结构与之前所学的线性存储结构有所差异，这缘于栈对数据 "存" 和 "取" 的过程有特殊的要求：

1. 栈只能从表的一端存取数据，另一端是封闭的，如图 1 所示；
2. 在栈中，无论是存数据还是取数据，都必须遵循"先进后出"的原则，即最先进栈的元素最后出栈。拿图 1 的栈来说，从图中数据的存储状态可判断出，元素 1 是最先进的栈。因此，当需要从栈中取出元素 1 时，根据"先进后出"的原则，需提前将元素 3 和元素 2 从栈中取出，然后才能成功取出元素 1。

因此，我们可以给栈下一个定义，即栈是一种只能从表的一端存取数据且遵循 "先进后出" 原则的线性存储结构。

通常，栈的开口端被称为栈顶；相应地，封口端被称为栈底。因此，栈顶元素指的就是距离栈顶最近的元素，拿图 2 来说，栈顶元素为元素 4；同理，栈底元素指的是位于栈最底部的元素，图 2 中的栈底元素为元素 1。

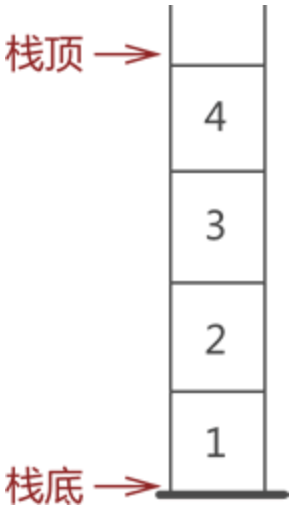


图 2 栈顶和栈底

## 进栈和出栈

基于栈结构的特点，在实际应用中，通常只会对栈执行以下两种操作：

- 向栈中添加元素，此过程被称为"进栈"（入栈或压栈）；
- 从栈中提取出指定元素，此过程被称为"出栈"（或弹栈）；

## 栈的具体实现

栈是一种 "特殊" 的线性存储结构，因此栈的具体实现有以下两种方式：

1. 顺序栈：采用顺序存储结构可以模拟栈存储数据的特点，从而实现栈存储结构；
2. 链栈：采用链式存储结构实现栈结构；

> 两种实现方式的区别，仅限于数据元素在实际物理空间上存放的相对位置，顺序栈底层采用的是数组，链栈底层采用的是链表。有关顺序栈和链栈的具体实现会在后续章节中作详细讲解。

## 栈的应用

基于栈结构对数据存取采用 "先进后出" 原则的特点，它可以用于实现很多功能。

例如，我们经常使用浏览器在各种网站上查找信息。假设先浏览的页面 A，然后关闭了页面 A 跳转到页面 B，随后又关闭页面 B 跳转到了页面 C。而此时，我们如果想重新回到页面 A，有两个选择：

- 重新搜索找到页面 A；
- 使用浏览器的"回退"功能。浏览器会先回退到页面 B，而后再回退到页面 A。

浏览器 "回退" 功能的实现，底层使用的就是栈存储结构。当你关闭页面 A 时，浏览器会将页面 A 入栈；同样，当你关闭页面 B 时，浏览器也会将 B入栈。因此，当你执行回退操作时，才会首先看到的是页面 B，然后是页面 A，这是栈中数据依次出栈的效果。

不仅如此，栈存储结构还可以帮我们检测代码中的括号匹配问题。多数编程语言都会用到括号（小括号、中括号和大括号），括号的错误使用（通常是丢右括号）会导致程序编译错误，而很多开发工具中都有检测代码是否有编辑错误的功能，其中就包含检测代码中的括号匹配问题，此功能的底层实现使用的就是栈结构。

同时，栈结构还可以实现数值的进制转换功能。例如，编写程序实现从十进制数自动转换成二进制数，就可以使用栈存储结构来实现。

以上也仅是栈应用领域的冰山一角，这里不再过多举例。在后续章节的学习中，我们会大量使用到栈结构。

接下来，我们学习如何实现顺序栈和链栈，以及对栈中元素进行入栈和出栈的操作。

上一节 下一节