**什么是数据结构？数据结构和数据类型的区别？**

https://www.zhihu.com/question/21165020/answer/118852959

其实严蔚敏编著的《数据结构》已经很好的回答了这个问题。

要理解这个问题先得具备以下三个基础概念，下方还会提及，这三个概念可以跳过不看：

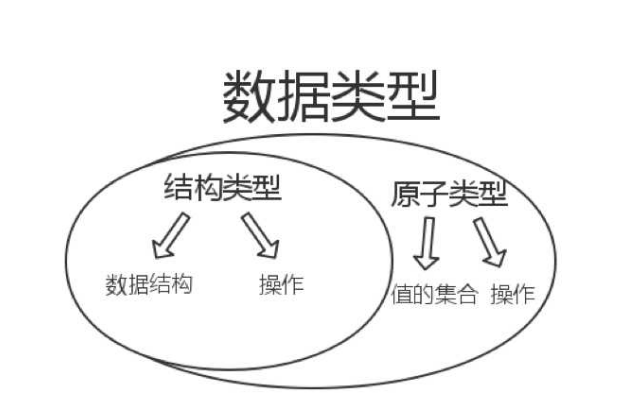
数据结构：是相互之间存在一种或多种特定关系的数据元素的集合，包括逻辑结构和物理结构。

数据类型：是一个值的集合和定义在这个值集上的一组操作的总称。

抽象数据类型：是指一个数学模型以及定义在该模型上的一组操作。

解读这三个基础概念，也就是它们之间的关系。

数据结构和数据类型其实是所属关系，是什么样的所属关系呢？看图：



解释一下这张我自己绘制的图：

一种数据结构＋定义在此种数据结构上的一组操作＝结构类型

一种值的集合＋定义在此种值的集合上的一组操作＝原子类型

结构类型＋原子类型＝数据类型

以下是背景知识，很重要，当然不想细看的直接看段尾的一句话总结。

三个基础概念：

数据结构：是相互之间存在一种或多种特定关系的数据元素的集合，包括逻辑结构和物理结构。

数据类型的定义是：是一个值的集合和定义在这个值集上的一组操作的总称。

数据类型的分类是：原子类型和结构类型。

融合2和3基础概念和上图可得：数据类型的两种类型是按照定义中两种“值的集合”来区分的。结构类型就是“值的集合”是一种数据结构，例如线性表、树和图；而原子类型就是“值的集合”是原子类型，例如C语言中的基本类型（整型、实型、字符型和枚举类型）、指针类型和空类型。

**一句话总结，数据结构是一种值的集合，这种值集＋值集上的操作就是结构类型，而结构类型是数据类型中的一种，所以数据结构属于数据类型。**

接下来解释数据类型和抽象数据类型的关系：

还是那句话，以下是背景知识，很重要，当然不想细看的直接看段尾的一句话总结。

两个基础概念：

数据类型：是一个值的集合和定义在这个值集上的一组操作的总称。

抽象数据类型：是指一个数学模型以及定义在该模型上的一组操作。

它们的异同其实就在字面上了——抽象。

抽象数据类型的定义仅取决于它的一组逻辑特性，而与其在计算机内部如何表示和实现无关，即无论其内部结构如何变化，只要它的数学特性不变，都不影响其外部的使用。”抽象“的意义在于数据类型的数学抽象特性。

一句话总结，数据类型和抽象数据类型的相同点在于它们具有相同的逻辑特性，不同点在于数据类型即关系数据集的逻辑特性又关系其物理特性，而抽象数据类型只关心数据集的抽象特性。

总结起来就是这三条基础概念：

数据结构：是相互之间存在一种或多种特定关系的数据元素的集合，包括逻辑结构和物理结构。

数据类型：是一个值的集合和定义在这个值集上的一组操作的总称。

抽象数据类型：是指一个数学模型以及定义在该模型上的一组操作。

以下是班门弄斧时间：

我认为，“数据类型”是计算机发展出来的一种重要思想，它将程序设计一分为二：实现基本数据类型和使用抽象数据类型。实现基本数据类型是指在物理层面实现数据类型的逻辑特性；使用抽象数据类型是指仅仅关注数据类型的逻辑特性，而不关注硬件实现。这样两边的人都可以更集中注意力于自己的工作范围，进行更深的钻研。