

# 定义

---

线性表(Linear List)是最常用且最简单的一种数据结构

简言之，线性表是 $n$ 个数据元素的有限序列；

$L = (A, B, C, \dots, Z)$ 是一个线性表；表中的元素是单个字母字符

在稍微复杂的线性表中，一个数据元素可以由若干个数据项(item)组成；

这种情况下，常把数据元素称为记录(record)，含有大量线性表的文件称为文件(file)

# 特点

---

线性表中的数据元素可以是多种多样的，但同一线性表中的元素必定具有相同的特性；即属同一数据对象，相邻数据元素之间存在着序偶关系；

# 前驱与后继

---

$(a_1, \dots, a_{i-1}, \dots, a_n)$

则表 $a_{i-1}$ 领先于 $a_i$ ， $a_i$ 领先于 $a_{i+1}$ ；

称 $a_{i-1}$ 是 $a_i$ 的直接前驱元素， $a_{i+1}$ 是 $a_i$ 的直接后继元素

当 $i = 1, 2, \dots, n-1$ 时， $a_i$ 有且仅有一个直接后继

当 $i = 2, 3, \dots, n$ 时， $a_i$ 有且仅有一个直接前驱

# 空表

---

线性表中元素的个数 $n(n \geq 0)$ 定义位线性表的长度， $n = 0$ 时空表；

# 抽象数据类型

---

```
ADT List {  
    数据对象:  $D = \{a_i | a_i \in \text{ElementSet}, i = 1, 2, \dots, n, n \geq 0\}$   
  
    数据关系:  $R1 = \{ \langle a_{i-1}, a_i \rangle | a_{i-1}, a_i \in D, i = 2, \dots, n \}$   
  
    基本操作:  
    //初始化线性表  
    Status InitList(ArrayList);
```

```
//销毁线性表
Status DestoryList(ArrayList);

//清空线性表
Status ClearList(ArrayList);

//是否为空线性表
Status ListEmpty(ArrayList);

//获取线性表长度
int ListLength(ArrayList);

//获取指定下标的元素
Element GetElement(ArrayList, index);

//定位具有指定值的元素的位置
int LocateElement(ArrayList, element);

//获取指定下标元素的前驱
Element PriorElement(ArrayList, index);

//获取指定下标元素的后继
Element NextElement(ArrayList, index);

//在指定位置插入一个元素
Status InsertElement(ArrayList, index, element);

//删除指定位置的元素
Status DeleteElement(ArrayList, index);

//遍历线性表
TraverseList(ArrayList);

}
```