

# 实验 1 日志

201808010515 黄茂荣

## 一. 实验设计

利用线性表实现对一串字符的处理，首先我们需要定义一个线性表的抽象数据类型（ADT），这个可以通过一个 List 的抽象类进行实现，这个类里面定义了数据类型，数据对象，以及相关操作（如清空，插入，删除，移动，查找，返回值等功能），相关操作通过函数来实现，为了便于同一个函数实现不同的功能，这里的函数需要加关键字 virtual 定义为纯虚函数。然后对于线性表的具体实现我们有两种方式，一种是通过顺序表（数组）来实现，另外一种是通过链表实现。实现线性表后，我们需要写主程序去测试线性表 ADT 的功能。

### 线性表的物理实现——链表

这里我们需要先定义一个结点类 link，用来表示链表中的结点，包括数据域 element 和一个地址指针 next，结点类中定义了带有参数的构造函数，用于后面对新结点的创建。LList 类公有继承抽象类 List，完成对线性表的基本操作函数的定义。

### 线性表的物理实现——顺序表

我们在顺序表的实现的时候，定义一个 AList 的类，公有继承 List，AList 内部定义线性表可容纳的最大元素个数 maxSize，当前元素个数 listSize，当前元素位置 curr，以及存放线性表元素的指针 \*listArray，List 类里还完成了对顺序表实现线性表的基本操作的函数定义。

## 二. 实验记录

2019 年 10 月 2 日

如何对线性表元素进行输出：最初的想法是调用函数 `getValue()` 对当前指针指向位置进行输出值，而每一次操作需要进行一次指针的移动，考虑到这个问题，修改函数，增添一个位置参数，直接对得到指定位置的函数值，再进行一次输出。

在表中插入元素：要注意使用链表时结点间指针的指向关系，比如在尾部插入元素的伪代码如下：  
`tail->next=new link<L>(it, NULL);`  
`tail=tail->next;`，需要新建结点，然后确定新结点的数据域和指针指向，再去新建结点间的连接关系。

2019 年 10 月 3 日

主程序测试的编写：首先定义一个线性表，确定其大小，利用 `for` 循环进行输入，调用 `append` 函数依次在表尾插入元素，测量线性表长度并输出，移动指针位置，插入元素，再打印出此时线性表中的元素，看测试的操作是否有问题，最后选择某个元素进行查找操作，检查查找功能是否有问题。这样实现了对线性表主要操作的简单测试，多试几组数据。

## 二. 心得和问题

2019 年 10 月 6-7 日

1. 写代码遇到的问题主要是对线性表的定义,在主程序中去定义一个线性表 `AList<int> L(100)`, 出现编译错误,之后发现是缺少相应的构造函数。当写过有参数的构造函数,程序就是自动匹配参数,构造函数需要根据程序的实现来写。另外一个问题就是,抽象类中定义纯虚函数后,在继承抽象类中的子类中没有对所有的纯虚函数进行定义,这样就会使得这个子类不能作为一种新的类型,导致编译不通过。

2. CG 上面题目对字符串进行处理,首先我们需要将输入的字符串存入线性表中,在输入单个字符的选择上需要注意输入终止条件,可以利用输入是否为 EOF 来判断是否结束,或者采用 `cin` 输入(这种输入不能处理含有空格的字符串)。