



中國石油大學 (华东)  
CHINA UNIVERSITY OF PETROLEUM

# 本科毕业设计(论文)

题 目：线性表的设计和实现实现实现实现实现

——这是副标题

学生姓名：张 三

学 号：1401013101

专业班级：电气工程及其自动化 14-5 班

指导教师：罗 翔

2020 年 6 月 14 日

# 线性表的设计和实现实现实现实现实现

——这是副标题

## 摘 要

结构算法设计和演示（C++）树和查找是在面向对象思想和技术的指导下，采用面向对象的编程语言（C++）和面向对象的编程工具（Borland C++ Builder 6.0）开发出来的小型应用程序。它的功能主要是将数据结构中链表、栈、队列、树、查找、图和排序部分的典型算法和数据结构用面向对象的方法封装成类，并通过类的对外接口和对象之间的消息传递来实现这些算法，同时利用 C++ Builder 6.0 中丰富的控件资源和系统资源对算法实现过程的流程和特性加以动态的演示，从而起到在数据结构教学中帮助理解、辅助教学和自我学习的作用。

**关键词：**数据结构；面向对象；可视化；算法；关键字 1；关键字 2；关键字 3；需要换行的关键字

# The design and implementation of the linear form

## Abstract

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

**Keywords:** Write Criterion; Typeset Format; Graduation Project (Thesis); Keyword One;  
Keyword Two; Keyword Newline

## 目 录

第 1 章 引言 .....	1
第 2 章 线性表的基本理论知识 .....	2
2.1 线性表的定义 .....	2
2.2 线性顺序表 .....	2
2.2.1 三级标题名 .....	2
2.2.2 三级标题名 .....	2
2.3 线性链表 .....	2
第 3 章 设计的主体内容 .....	3
3.1 系统结构的设计 .....	3
3.2 交互界面的设计和实现 .....	3
3.3 线性表的 OOP 序设计 .....	3
3.3.1 线性表的顺序存储的实现 .....	3
3.3.2 线性表的链表存储的实现 .....	4
第 4 章 实验及结果分析 .....	5
第 5 章 结论 .....	6
致谢 .....	7
参考文献 .....	8
附录 .....	9
A 名词术语及缩略词 .....	9
A.1 Some Appendix .....	9
B Appendix 2 .....	10
B.1 Some Other Appendix .....	10

## 第 1 章 引言

计算机与网络技术的高速发展，特别是面向对象技术的出现，使得 C++ 的开发得到了迅速普及。

本课题主要……………

## 第2章 线性表的基本理论知识

### 2.1 线性表的定义

线性表是最简单、最常用<sup>[1]</sup>的一种数据结构。线性表<sup>[2]</sup>是  $n$  ( $n \geq 0$ ) 个数据元素的有限序列。

……。

### 2.2 线性顺序表

线性表的顺序存储结构的特点是为表中相邻的元素  $a_i$  和  $a_{i+1}$  赋以相邻的存储位置。

#### 2.2.1 三级标题名

#### 2.2.2 三级标题名

(1) 三级以下标题

### 2.3 线性链表

线性表的链式存储结构的特点是用一组任意的存储单元存储线性表的数据元素（这组元素可以是连续的，也可以是不连续的）。

**定义 2.1.** 线性表的定义为

**定理 2.1.** 设  $sxibaxbxugbuaxbsusabx$

**证明.** 由线性表的特殊定义可得。 □

设  $sxibaxbxugbuaxbsusabx$

**引理 2.2.** 设  $sxibaxbxugbuaxbsusabx$

**证明.** 由线性表的特殊定义可得。 □

## 第3章 设计的主体内容

在着手进行上机设计之前首先做好大量准备: 应熟悉课题, 进行调查研究, 收集国内、外资料、分析研究; 交互界面的设计和实现。

……。

### 3.1 系统结构的设计

……。

### 3.2 交互界面的设计和实现

交互界面的设计应遵循……。

$$b \approx \frac{L_0}{\rho \tan(\theta_0) + z_0} \quad (3-1)$$

式中,  $z_0$  为 *Goos-Hanchen* 位移;  $\theta_0$  为光波的入射角。

由公式 (3-1) 可以看出……。

### 3.3 线性表的 OOP 序设计

计算机内部可以采用两种不同方法来表示一个线性表, 它们分别是顺序表示法和链表表示法。

……。

过阻尼响应如图3-1所示。

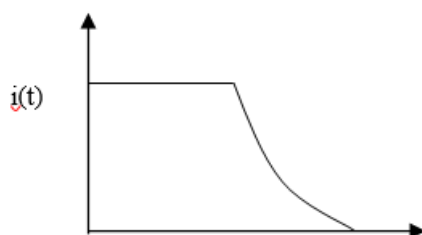


图 3-1 过阻尼响应

#### 3.3.1 线性表的顺序存储的实现

……

以上是顺序表的实现过程, 第 1-16 行包含了 list 类的说明, 接下来是成员函数的定义。

……。

### 3.3.2 线性表的链表存储的实现

.....

链表的实现包括两个类定义，第一个是 `link` 类，第二个是 `list` 类。由于一个链表由若干个单独的链结点对象组成，因此一个链结点应当作为单独的 `link` 类实现。

.....

.....



## 第4章 实验及结果分析

例如由于起初未能真正掌握各种控件的功能，我设想是要一个下拉菜单，但是学识肤浅的我试了很多种就是达不到我要的效果，……。

……

关于……的影响如表4-1所示。

……

表 4-1 激光入射功率密度对导轨滚道表面硬化层深和显微硬度的影响

试验编号	功率密度	辐照时间	显微硬度	硬化层深
t-1	6.37×103	0.067	570, 456	0.354
t-2	6.37×103	0.067	570, 456	0.354
t-3	6.37×103	0.067	570, 456	0.354
t-4	6.37×103	0.067	570, 456	0.354
t-5	6.37×103	0.067	570, 456	0.354

鉴于表格复杂性，此处提供了可换行示例表见表4-2

表 4-2 可换行示例表

1	2	3
1	3	6
1	3	6
2	4444444444	6
	5555555555	

此处也提供了多列合并示例表如表4-3

表 4-3 多列合并示例表

ZZ		XX						
		CC		VV				
				BB			NN	
	A	S	D	F	G	H	J	K
Q	√	√						
T			√	√			√	
Y			√	√				√

## 第 5 章 结论

本课题采用 C++ 语言、面向对象的设计方法实现数据结构的重要算法。

.....

而且还存在着许多不足之处。如：

## 致 谢

大学四年的学习生活即将结束，在此，我要感谢所有曾经教导过我的老师和关心过我的同学，他们在我成长过程中给予了我很大的帮助。本文能够成功的完成，要特别感谢我的导师 XXX 教授的关怀和教导。

.....

## 参考文献

[1] 按照格式要求填写这里

[2]

[3]

## 附 录

### A 名词术语及缩略词

#### A.1 Some Appendix

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

## **B Appendix 2**

### **B.1 Some Other Appendix**