

3.2 链栈

链栈是操作受限的单链表。链栈的结点定义与单链表的结点定义一样，但取名为 SNode。链栈数据元素插入与删除限定在栈顶进行，表头处为栈顶；因不能对栈中的第 i 个元素进行操作，这使得链栈的基本操作的个数少于单链表，操作也简单。

一、程序设计简介

本验证程序包括两个文件 LinkStack.h 和 LinkStack.cpp。

(1) 头文件 LinkStack.h，其中包括 2 个内容：

① 链栈的结点 SNode 定义

② 链栈的基本操作实现，包括：初始化链栈 InitStack() (算法 3.6)、销毁栈 DestroyStack() (算法 3.7)、数据元素入栈 Push() (算法 3.8)、数据元素出栈 Pop() (算法 3.9)、取栈顶元素 GetTop() (算法 3.10)、清空栈 StackEmpty()、测栈空 StackEmpty()、显示栈 DispStack() 等。

(2) 源程序文件 LinkStack.cpp，通过调用 LinkStack.h 中定义的操作实现程序的各项功能。程序功能结构如图 1.3.3 示。

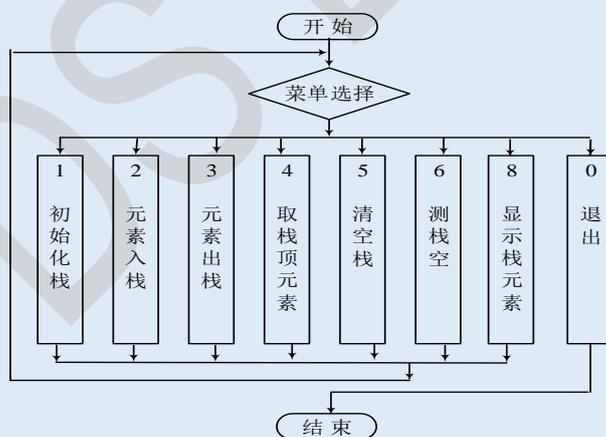


图1.3.1 链栈验证实验程序结构框图

二、运行说明

运行程序，显示如图 1.3.4 所示界面。

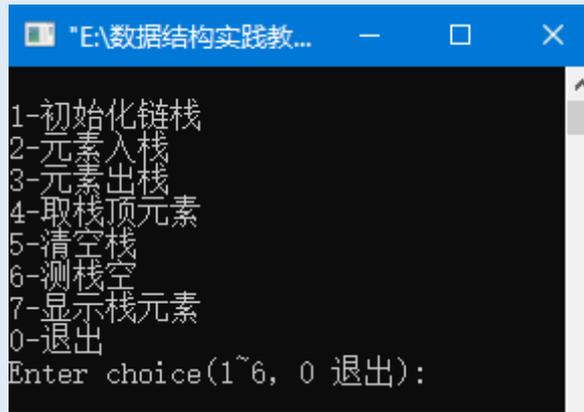


图1.3.2 链栈程序运行界面

case 1: 键入“1”，选择功能 1，初始化栈

1.1 屏幕显示创建成功信息提示。

注：用功能 6 测栈是否为空

case 2: 键入“2”，选择功能 2，元素入栈

2.1 按屏幕提示，输入入栈元素值（1 个整数）。

2.2 若入栈成功，屏幕显示元素入栈后栈中所有元素；否则显示入栈不成功提示信息。

case 3: 键入“3”，选择功能 3，元素出栈

3.1 屏幕显示出栈元素。

3.2 若出栈成功，屏幕显示元素出栈后栈中所有元素；否则显示出栈不成功提示信息。

case 4: 键入“4”，选择功能 4，取栈顶元素

栈非空，屏幕显示栈顶元素；否则显示不成功“栈空”提示信息。

case 5: 键入“5”，选择功能 5，清空栈

5.1 屏幕显示测试结果。

5.2 用功能 7 查看前后栈中元素；用功能 6 测栈空，用功能 7 查看栈中元素，验证的正确性。

case 6: 键入“6”，选择功能 6，测栈空

6.1 屏幕显示测试结果。

6.2 用功能 7 查看栈中元素，验证的正确性。

case 7: 键入“7”，选择功能 7，显示栈中元素

屏幕显示栈中所有元素。

case 0: 键入“0”，选择功能 0，程序退出

屏幕显示“结束运行 bye-bye!”，按任意键，结束程序运行。

三、思考题

1. 研读源程序，回答下列问题

- (1) 链栈采用了什么形式的链表？
- (2) 验证程序中栈元素类型是什么？如果为其它类型，如何修改程序？
- (3) 验证程序实现了哪些栈的基本操作？对应的函数分别是哪一个？
- (4) 有清空栈对应的函数吗？清空栈是如何实现的？
- (5) 栈顶指针指向哪里？GetTop() 操作栈顶指针动了么？
- (6) 元素入栈时，结点插入在链表的哪个地方？
- (7) 出栈时删除是链表的哪个结点？

2. 运行程序，回答下列问题

- (8) 元素 11、22、33、44 依次入栈，操作 7 显示栈元素，看到的序列应该是什么？
- (9) 对于空栈，执行功能 3、4、6、7，理解与分析运行结果。
- (10) 程序启动成功后直接按“0”退出，可以正常退出吗？为什么？