



链滴

数据结构——栈的实现及应用

作者: [Dronsure](#)

原文链接: <https://ld246.com/article/1453957954840>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)


```

#ifndef STACK_H<br>
#define STACK_H<br>
typedef struct Node<br>
{<br>
ElementType data;<br>
struct Node *next;<br>
struct Node *previous;<br>
} <em>Stack, StackNode; //数据结点<br>
Stack CreateStack(); //创建一个栈<br>
char isEmpty(Stack stackIns); //判断栈是否为空<br>
ElementType pop(Stack stackIns); //出栈<br>
void push(Stack stackIns, ElementType dataSource); //入栈<br>
#endif </em> STACK_H *</p>
<p>*<br>
*File Name: Stack.c<br>
*Author Name: Dronsure<br>
*Create Time: Fri Sep 18 21:03:17 2015<br>
<em><br>
#include "Stack.h"<br>
#include &lt;malloc.h&gt;<br>
Stack CreateStack()<br>
{<br>
Stack head = NULL;<br>
head = (StackNode</em>)malloc(sizeof(StackNode));<br>
head-&gt;next = NULL;<br>
return head;<br>
}<br>
char isEmpty(Stack stackIns)<br>
{<br>
return stackIns-&gt;next == NULL;<br>
}<br>
void push(Stack stackIns, ElementType dataSource)<br>
{<br>
StackNode *p = stackIns;<br>
StackNode *tmp;<br>
while(p-&gt;next != NULL)<br>
p = p-&gt;next;<br>
tmp = (StackNode <em>) malloc(sizeof(StackNode));<br>
VolutionElementTypeData(&tmp-&gt;data,dataSource);<br>
tmp-&gt;next = NULL;<br>
tmp-&gt;previous = p;<br>
p-&gt;next = tmp;<br>
}<br>
ElementType pop(Stack stackIns)<br>
{<br>
StackNode</em> p = stackIns;<br>
ElementType tmp;<br>
while(p-&gt;next != NULL)<br>
p = p-&gt;next;<br>
p-&gt;previous-&gt;next = NULL;<br>
VolutionElementTypeData(&tmp, p-&gt;data);<br>
free(p);<br>
return tmp;<br>
}</p>

```

```
<p>/*<br>
*File Name: main.c<br>
*Author Name: Dronsure<br>
*Create Time: Fri Sep 18 20:46:49 2015<br>
<em>/<br>
#include &lt;stdio.h&gt;<br>
#include "Stack.h"<br>
int main(int argc, char</em> argv[])<br>
{<br>
Stack stackIns;<br>
stackIns = CreateStack();<br>
printf("%d\n",isEmpty(stackIns));<br>
push(stackIns, InputElementTypeData());<br>
printf("%d\n",isEmpty(stackIns));<br>
push(stackIns, InputElementTypeData());<br>
PrintElementTypeData(pop(stackIns));<br>
PrintElementTypeData(pop(stackIns));<br>
printf("%d\n",isEmpty(stackIns));<br>
return 0;<br>
}</p></pre><p></p>
<p>&nbsp;</p>
```