

C\_C 中结构体(struct)知识点强化 (一) PDF转换可能丢失图片或格式, 建议阅读原文

[https://www.100test.com/kao\\_ti2020/136/2021\\_2022\\_C\\_C\\_\\_\\_E4\\_B8\\_AD\\_E7\\_BB\\_c97\\_136118.htm](https://www.100test.com/kao_ti2020/136/2021_2022_C_C___E4_B8_AD_E7_BB_c97_136118.htm) 为了进一部的学习结构体这一重要的知识点, 我们今天来学习一下链表结构。结构体可以看做是一种自定义的数据类型, 它还有一个很重要的特性, 就是结构体可以相互嵌套使用, 但也是有条件的, 结构体可以包含结构体指针, 但绝对不能在结构体中包含结构体变量。 struct test { char name[10]. float socre. test \*next. }//这样是正确的! struct test { char name[10]. float socre. test next. }//这样是错误的! 利用结构体的这点特殊特性, 我们就可以自己生成一个环环相套的一种射线结构, 一个指向另一个。来源

: [www.examda.com](http://www.examda.com) 链表的学习不像想象的那么那么容易, 很多人学习到这里的时候都会碰到困难, 很多人也因此而放弃了学习, 在这里我说, 一定不能放弃, 对应它的学习我们要进行分解式学习, 方法很重要, 理解需要时间, 不必要把自己逼迫的那么紧, 学习前你也得做一些最基本的准备工作, 你必须具备对堆内存的基本知识的了解, 还有就是对结构体的基本认识, 有了这两个重要的条件, 再进行分解式学习就可以比较轻松的掌握这一节内容的难点。下面我们给出一个完整的创建链表的程序, 不管看的懂看不懂希望读者先认真看一下, 想一想, 看不懂没有关系, 因为我下面会有分解式的教程, 但之前的基本思考一定要做, 要不即使我分解了你也是无从理解的。代码如下, 我在重要部分做了注解:

```
#include using namespace std. struct test { char name[10]. float socre. test *next. }. test *head.//创建一个全局的引导进入链表的
```

```
指针 test *create() { test *ls;//节点指针 test *le;//链尾指针 ls = new
test;//把ls指向动态开辟的堆内存地址
cin>>ls->name>>ls->score. head=NULL;//进入的时候先不设置head指针指向任何地址,因为不知道是否一上来就输入null跳出程序 le=ls;//把链尾指针设置成刚刚动态开辟的堆内存地址,用于等下设置le->next,也就是下一个节点的位置
while(strcmp(ls->name,"null")!=0)//创建循环条件为ls->name的值不是null,用于循环添加节点 { if(head==NULL)//判断是否是第一次进入循环 { head=ls;//如果是第一次进入循环,那么把引导进入链表的指针指向第一次动态开辟的堆内存地址 } else { le->next=ls;//如果不是第一次进入那么就把上一次的链尾指针的le->next指向上一次循环结束前动态创建的堆内存地址 }
100Test 下载频道开通 , 各类考试题目直接下载。详细请访问
www.100test.com
```