设计二叉链表结构的相关函数库 - 百度文库

请稍等...

百度文库

搜索文档或关键词

普通分享 >

设计二叉链表结构的相关函数库

共享文档

2011-12-25

1页

用App免费查看

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 课程设计报告 |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 课程： | 数据结构 |  |
| 学号： |  |  |
| 姓名： |  |  |
| 班级： |  |  |
| 教师： |  |  |
| 时间： | 20 11 . 12 . 16 ～ 20 11 . 12 . 30 |  |
|  |  |  |
|  |  |  |
|  |  |  |
| 计算机科学与技术系 |  |  |
|  |  |  |

|  |
| --- |
| 设计名称： 设计二叉链表结构的相关函数库 |
| 设计目的与要求：  使用 Microsoft Visual C++ 设计二叉链表结构的相关函数库，以便在程序设计中调用  设计二叉链表结构的相关函数库，在程序设计中调用，并实现二叉树的各种基本函数以及常用函数。 |
| 设计 所用软件 环境：  Microsoft Visual C++ 6.0 |
| 问题的模型化描述及求解算法的简要描述：    1. 数据结构  typedef char datatype;  typedef struct node // 定义二叉树结点类型  {  datatype data;  struct node \*lchild;  struct node \*rchild;  }Btnode,\* Btree;  typedef struct QueueNode{  BiTree data;  struct QueueNode \*next;  } LinkQueueNode;// 队列中的每个节点  typedef struct  {  LinkQueueNode \*front;  LinkQueueNode \*rear;  } LinkQueue;// 队列  2. 模块划分  1. 入队  void EnterQueue(LinkQueue \*Q,BiTree x)  2. 队列判空  int QueueIsEmpty(LinkQueue \*Q)  3. 出队  void DeleteQueue(LinkQueue \*Q,BiTree \*x)  4. 根据先序递归建立二叉树  Btree pre\_creat()  5. 递归遍历输出函数  void preorder\_btree(Btree root) // 由先根序列遍历输出二叉树  void inorder\_btree(Btree root) // 由中根序列遍历输出二叉树  void postorder\_btree(Btree root) // 由后根序列遍历输出二叉树  6. 层次遍历输出算法  void level\_btree(Btree root)  7. 求二叉树深度得算法  void depth(BiTree root,  8. 求二叉树叶子结点数的算法  void countleaf(BiTree root |
| 软件组成及使用说明：  软件组成 ：  Visual C++6.0 由许多组件组成，包括编辑器、调试器以及程序向导 AppWizard 、类向导 Class Wizard 等开发工具。  使用说明：  (1) 编辑 ( 把程序代码输入，交给计算机 ) 。  (2) 编译 ( 成目标程序文件 .obj) 。编译就是把高级语言变成计算机可以识别的 2 进制语言，计算机只认识 1 和 0 ，编译程序把人们熟悉的语言换成 2 进制的。编译程序把一个源程序翻译成目标程序的工作过程分为五个阶段：词法分析；语法分析；语义检查和中间代码生成；代码优化；目标代码生成。主要是进行词法分析和语法分析，又称为源程序分析，分析过程中发现有语法错误，给出提示信息。  (3) 链接 ( 成可执行程序文件 .exe) 。链接是将编译产生的 .obj 文件和系统库连接装配成一个可以执行的程序。由于在实际操作中可以直接点击 Build 从源程序产生可执行程序，可能有人就会置疑：为何要将源程序翻译成可执行文件的过程分为编译和链接两个独立的步骤，不是多此一举吗？之所以这样做，主要是因为：在一个较大的复杂项目中，有很多人共同完成一个项目 ( 每个人可能承担其中一部分模块 ) ，其中有的模块可能是用汇编语言写的，有的模块可能是用 VC 写的，有的模块可能是用 VB 写的，有的模块可能是购买 ( 不是源程序模块而是目标代码 ) 或已有的标准库模块，因此，各类源程序都需要先各自编译成目标程序文件 (2 进行机器指令代码 ) ，再通过链接程序将这些目标程序文件连接装配成可执行文件。  (4) 运行 ( 可执行程序文件 ) 。  上述四个步骤中，其中第一步的编辑工作是最繁杂而又必须细致地由人工在计算机上来完成，其余几个步骤则相对简单，基本上由计算机来自动完成。 |
| 程序清单：  #include<stdio.h>  #include<stdlib.h>  typedef struct Node{  char data;  struct Node \* Lchild;  struct Node \* Rchild;  } BiTNode,\*BiTree;// 二叉树节点，二叉链表  typedef struct QueueNode{  BiTree data;  struct QueueNode \*next;  } LinkQueueNode;// 队列中的每个节点  typedef struct  {  LinkQueueNode \*front;  LinkQueueNode \*rear;  } LinkQueue;// 队列  /\* 队列的初始化 \*/  void InitQueue(LinkQueue \*Q)  {  Q->front = (LinkQueueNode \*)malloc(sizeof(LinkQueueNode));  if(Q->front != NULL){  Q->rear = Q->front;  Q->front->next = NULL;  } else printf(" 分配空间失败 !\n");  }  /\* 入队 \*/  void EnterQueue(LinkQueue \*Q,BiTree x)  {  LinkQueueNode \*NewNode;  NewNode = (LinkQueueNode \*)malloc(sizeof(LinkQueueNode));  if(NewNode != NULL){  NewNode->data = x;  NewNode->next = NULL;  Q->rear->next = NewNode;  Q->rear = NewNode;  }  }  /\* 队列判空 \*/  int QueueIsEmpty(LinkQueue \*Q)  {  if(Q->front == Q->rear)  return 1;  else return 0;  }  /\* 出队 \*/  void DeleteQueue(LinkQueue \*Q,BiTree \*x)  {  LinkQueueNode \*p;  if(Q->front == Q->rear)  return ;  p= Q->front->next;  Q->front->next = p->next;  if(Q->rear == p)  Q->rear = Q->front;  \*x = p->data;  free(p);  }  /\* 利用扩展先序遍历序列  　　创建二叉链表 \*/  void CreateBiTree(BiTree \*bt)  {  char ch;  ch = getchar();  if(ch == '.') \*bt = NULL;  else  {  \*bt = (BiTree)malloc (sizeof(BiTNode));  (\*bt)->data = ch;  CreateBiTree(&((\*bt)->Lchild));  CreateBiTree(&((\*bt)->Rchild));  }  }  /\* 先序递归遍历二叉树 \*/  void PreOrder(BiTree root)  {  if(root != NULL){  printf("%c ",root->data);  PreOrder(root->Lchild);  PreOrder(root->Rchild);  }  }  /\* 后序递归遍历二叉树 \*/  void PostOrder(BiTree root)  {  if(root != NULL){  PostOrder(root -> Lchild);  PostOrder(root -> Rchild);  printf("%c ",root->data);  }  }  void InOrder(BiTree root)  {  if(root != NULL){  InOrder(root->Lchild);  printf("%c ",root->data);  InOrder(root->Rchild);  }  }  /\* 层序遍历  　　 对给定的二叉树进行层序遍历 \*/  void LayerOrder(BiTree root)  {  BiTree \*x;  // 这里要记得申请空间  x = (BiTree \*)malloc(sizeof(BiTree));  if(x == NULL){  printf(" 内存分配失败 !\n");  }  LinkQueue \*Q;  Q = (LinkQueue \*)malloc(sizeof(LinkQueue));  InitQueue(Q);  EnterQueue(Q,root);  while(!QueueIsEmpty(Q)){  DeleteQueue(Q,x);  printf("%c ",(\*x)->data);  if((\*x)->Lchild)EnterQueue(Q,(\*x)->Lchild);  if((\*x)->Rchild)EnterQueue(Q,(\*x)->Rchild);  }  }  void countleaf(BiTree root,int&n)  { if (root)  {  countleaf(root->Lchild,n);  if(!root->Lchild && !root->Rchild) n++;  countleaf(root->Rchild,n);  }  }  void depth(BiTree root,int &dep)  { int dep1,dep2;  if(!root) dep=0;  else  {depth(root->Lchild,dep1);  depth(root->Rchild,dep2);  dep=dep1>dep2?dep1+1:dep2+1;  }  }  int main(int argc , char \*\*argv)  {  int n=0,dep;  BiTree root;  CreateBiTree(&root);  printf(" 先序递归遍历 :\n");  PreOrder(root);  printf("\n");  printf(" 中序递归遍历 :\n");  InOrder(root);  printf("\n");  printf(" 后序递归遍历 :\n");  PostOrder(root);  printf("\n");  printf(" 层序遍历 :\n");  LayerOrder(root);  printf("\n");  depth(root,dep);  printf(" 深度 dep=%d\n",dep);  countleaf(root,n);  printf(" 叶子结点数 n=%d\n",n);  printf("\n");  return 0;  } |
| 程序运行结果： |
| 设计总结和体会：  本次课程设计实现了二叉链表的相关函数库的调用。为了实现以链表为存储结构的二叉树的有关操作，要熟练掌握二叉链表的特性，但对于一些算法较为复杂，代码量多些，容易出现一些变量的定义、函数声明、函数调用等细节上的问题出错。在本程序的设计过程中，为了克服以上困难，采取了一些措施：建立清晰的程序设计的步骤方法，分步各个模块程序设计，进行仔细的总体结构设计，反复调试、细心观察达到完善整个系统等。  二叉树的递归算法主要是将二叉树存储到链表结构中。遍历是二叉树各种操作的基础，先序、中序、后序是二叉树遍历的三种基本遍历方法。而这些都是数据结构的基础内容，是我们必须理解和牢记的基础知识。将这些基础算法综合起来，更能清晰地认识和理解各种算法的作用。当然，要学会编程不会仅局限于课本知识，而是根据课本知识进行有效的拓展，并且不得不学会在众多的参考资料中搜索有用的自己所需的知识，并迫使自己去学习掌握它们，从中不断提高自己。虽然程序规模不大，我们依然为此付出了努力，仍免不了各种错误的出现。编程过程需要很大的毅力和耐心，而且要有良好的思维和扎实的专业基础知识，所以我们需要不断的学习，发现自身不足之处并改正它，逐步提高自身能力，不断取得进步。对于数据结构的学习，一直感到很吃力，也想过放弃。通过实践，让我们认识到知识的运用性，并加深对基础知识的理解，从中了解自己需要学习的东西并学会自学。在此我们要感 谢我的 老师对我们专心致志的辅导，让我们学会了许多分析和解决问题的方法，让我们受益匪浅。今后我们将加倍努力学习专业知识，为自己从事的职业打下坚实基础。 |
| 设计成绩： 教师签名：  年 月 日 |
|  |

点击加载更多

金榜VIP已享免费阅读及下载

加载失败，请重试

打开百度APP阅读全文

立即领取

VIP教育大礼包

热门小说免费读

本文配套内容

含${item.docNum}篇文档

${item.title}

￥**${item.price}**

立即购买

查看文集

### 精品课程

* ${item.title}
* 免费 ￥${item.price}￥${item.oriPrice} ￥${item.oriPrice} ${item.orgName}
* ${item.videoCount}课节

相关推荐文档

* ${searchSpecial.title}
* ${v.docTitle}
* *推荐* *热门* *好评*
* 用App查看
* 打开百度APP

返回百度搜索

下载原文档，方便随时阅读

下载文档

## 2亿文档资料库

涵盖各行课件、资料、模板、题库、报告等

## 多种记录存储好工具

提供图转文字、拍照翻译、语音速记等

## APP端内容永久保存

随时阅读，多端同步

立即下载

看视频广告，获取20元代金券礼包

看视频，立领券 视频大小约3.7M

您是老用户，送您2张代金券

* 5元
* 适用除连续包月外的其他VIP
* 24小时内有效
* 10元
* 限百度文库VIP-12个月适用
* 24小时内有效

领取优惠券

您已成功领取老用户福利

已转存到百度网盘

存储在文件夹【来自：百度文库】

去看看

文库新人专享礼包

限时免费

价值¥500+

去文库APP免费领