**《数据结构程序设计》**

**实验报告**

专 业： 空间信息与数字技术

班 级： 18级空间信息与数字技术

学号姓名： 刘姝麟

指导教师： 袁 莹

成 绩：

**计算机与信息工程学院**

**2019 年 10月**

**代码实现：**

1. **树的实现**

**头文件：**

#pragma once

#ifndef Tree\_H

#define Tree\_H

const int Max = 20;

struct TNode

{

char data;

TNode\* firstchild, \* rightsib;

};

class Tree

{

public:

Tree();

~Tree() { Release(root); }

void PreOrder() { PreOrder(root); }

void PostOrder() { PostOrder(root); }

private:

TNode\* root;

TNode\* Creat(TNode\* bt);

void Release(TNode\* bt);

void PreOrder(TNode\* bt);

void PostOrder(TNode\* bt);

};

#endif // !Tree\_H

**源文件：**

（1）

#include <iostream>

using namespace std;

#include "Tree.h"

Tree::Tree()

{

TNode\* Q[Max] = { NULL };

int front = -1, rear = -1;

char ch1 = '#', ch2 = '#';

TNode\* p = NULL, \* q = NULL;

cout << "请输入根结点：";

cin >> ch1;

p = new TNode; p->data = ch1;

p->firstchild = p->rightsib = NULL;

root = p;

Q[++rear] = p;

cout << "请输入结点对，以空格分隔：";

fflush(stdin);

ch1 = getchar(); getchar(); ch2 = getchar();

while (ch1 != '#' || ch2 != '#')

{

p = new TNode; p->data = ch2;

p->firstchild = p->rightsib = NULL;

Q[++rear] = p;

while (front != rear)

{

q = Q[front + 1];

if (q->data != ch1)

front++;

else

{

if (q->firstchild == NULL)

q->firstchild = p;

else

{

while (q->rightsib != NULL)

q = q->rightsib;

q->rightsib = p;

}

break;

}

}

cout << "请输入结点对，以空格分隔：";

fflush(stdin);

ch1 = getchar(); getchar(); ch2 = getchar();

}

}

void Tree::Release(TNode\* bt)

{

if (bt == NULL)return;

else

{

Release(bt->firstchild);

Release(bt->rightsib);

delete bt;

}

}

void Tree::PreOrder(TNode\* bt)

{

if (bt == NULL)return;

else

{

cout << bt->data;

PreOrder(bt->firstchild);

PreOrder(bt->rightsib);

}

}

void Tree::PostOrder(TNode\* bt)

{

if (bt == NULL)return;

else

{

PostOrder(bt->firstchild);

PostOrder(bt->rightsib);

cout << bt->data;

}

(2)

#include <iostream>

using namespace std;

#include "Tree.h"

int main()

{

Tree t1;

t1.PreOrder();

cout << endl;

t1.PostOrder();

cout << endl;

return 0;

}

**2、二叉树的实现**

**主函数：**

#pragma once

#ifndef BiTree\_H

#define BiTree\_H

struct BiNode

{

char data;

BiNode\* lchild, \* rchild;

};

class BiTree

{

public:

BiTree() { root = Creat(root); }

~BiTree() { Release(root); }

void PreOrder() { PreOrder(root); }

void InOrder() { InOrder(root); }

void PostOrder() { PostOrder(root); }

private:

BiNode\* root;

BiNode\* Creat(BiNode\* bt);

void Release(BiNode\* bt);

void PreOrder(BiNode\* bt);

void InOrder(BiNode\* bt);

void PostOrder(BiNode\* bt);

};

#endif // !BiTree\_H

**原函数**

#include <iostream>

using namespace std;

#include"Bitree.h"

BiNode\* BiTree::Creat(BiNode\* bt)

{

char ch;

cout << "请输入创建一棵二叉树的结点数据" << endl;

cin >> ch;

if (ch == '#')return NULL;

else

{

bt = new BiNode;

bt->data = ch;

bt->lchild = Creat(bt->lchild);

bt->rchild = Creat(bt->rchild);

}

return bt;

}

void BiTree::Release(BiNode\* bt)

{

if (bt != NULL)

{

Release(bt->lchild);

Release(bt->rchild);

delete bt;

}

}

void BiTree::PreOrder(BiNode\* bt)

{

if (bt == NULL)return;

else

{

cout << bt->data << " ";

PreOrder(bt->lchild);

PreOrder(bt->rchild);

}

}

void BiTree::InOrder(BiNode\* bt)

{

if (bt == NULL)return;

else

{

InOrder(bt->lchild);

cout << bt->data << " ";

InOrder(bt->rchild);

}

}

void BiTree::PostOrder(BiNode\* bt)

{

if (bt == NULL)return;

else

{

PostOrder(bt->lchild);

PostOrder(bt->rchild);

cout << bt->data << " ";

}

}

**2、**#include<iostream>

using namespace std;

#include "Bitree.h"

int main()

{

BiTree T;

cout << "------前序遍历------" << endl;

T.PreOrder();

cout << endl;

cout << "------中序遍历------" << endl;

T.InOrder();

cout << endl;

cout << "------后序遍历------" << endl;

T.PostOrder();

cout << endl;

return 0;

}

1. **心得体会：**

**本次的数据结构实验是关于二叉树的实现和树的实现两个方面。主要是进一步锻炼了我对代码的熟知程度，能够更加快速准确的输入代码。另外则是加深了我对这两个实验的印象。有助于我更好的学习数据结构实验教程这块内容。**