**Лабораторная работа №10. Бинарные деревья**

**Вариант 10**

Условие:

Написать функцию, которая записывает в файл G элементы дерева поиска Т в порядке их возрастания.

Код:

#include "stdafx.h"

#include <iostream>

#include "fstream"

#pragma warning(disable:4703)

#define TRUE 1

#define FALSE 0

int sch;

using namespace std;

struct node

{ int Key;

int Count;

node \*Left;

node \*Right;

};

node \*Tree; //Указатель на корень дерева

node \*Res; //Указатель на найденную вершину

int B; //Признак нахождения вершины в дереве

fstream file;

node\*\* GetTree()

{

return &Tree;

}

void Search (int x, node \*\*p) //Поиск звена x

{ if (\*p == NULL) //\*p - указатель на вершину

{ \*p = new(node);

(\*\*p).Key = x;

(\*\*p).Count = 1;

(\*\*p).Left = (\*\*p).Right = NULL;

}

else if(x<(\*\*p).Key)

Search (x, &((\*\*p).Left));

else if (x>(\*\*p).Key) Search (x, &((\*\*p).Right));

else (\*\*p).Count += 1;

}

void BuildTree () //Построение бинарного дерева

{ int el;

cout<<"Введите ключи вершин дерева: (если закончить - 0)\n"; cin>>el;

while (el != 0)

{ Search (el, &Tree); cin>>el; }

}

void Vyvod (node \*\*w, int l) //Вывод на экран

{ int i;

if (\*w != NULL) //\*w - указатель на корень

{ Vyvod (&((\*\*w).Right), l + 1);

for (i = 1; i <= l; i++) cout<<" ";

cout<<(\*\*w).Key<<endl;

Vyvod (&((\*\*w).Left), l + 1);

}

}

int Poisk (int k) //Поиск вершины с ключом k

{ node \*p,\*q; B = FALSE; p = Tree;

if (Tree != NULL)

do

{ q = p;

if ((\*p).Key == k) B = TRUE;

else { q = p; if (k < (\*p).Key) p = (\*p).Left; else p = (\*p).Right; }

} while (!B && p != NULL);

Res = q;

return B;

}

void Addition (int k) //Добавление звена k

{ node \*s; Poisk (k);

if (!B)

{ s = new(node);

(\*s).Key = k; (\*s).Count = 1;

(\*s).Left = (\*s).Right = NULL;

if (Tree == NULL) Tree = s;

else if (k<(\*Res).Key) (\*Res).Left = s;

else (\*Res).Right = s;

}

else (\*Res).Count += 1;

}

void obxod(node\*e)

{

if(e!=NULL)

{

obxod(e->Left);

file<<e->Key<<" ";

obxod(e->Right);

}

}

void main ()

{ setlocale (LC\_CTYPE, "Russian");

int el;

do{

cout<<"1-Создать дерево\n";

cout<<"2-Поиск вершины в дереве\n";

cout<<"3-Ввод ключа добавляемой вершины\n";

cout<<"4-Запись в файл в порядке возрастания\n";

cout<<"0-Выход\n";

cin>>sch;

switch (sch)

{

case 0:break;

case 1:BuildTree ();

Vyvod(&Tree, 0);

break;

case 2: cout<<"\nВведите вершину которую нужно найти: ";

cin>>el;

if (Poisk (el)) cout<<"В дереве есть такая вершина!\n";

else cout<<"В дереве нет такой вершины!\n";

break;

case 3: cout<<"Введите ключ добавляемой вершины: ";

cin>>el;

Addition (el);

Vyvod (GetTree(), 0);

break;

case 4:file.open("file.txt");

obxod(Tree);

file.close();break;

}

}while(sch!=0);

}

Cкриншот:

