***Министерство образования Республики Беларусь***

***Учреждение Образования***

***«Брестский Государственный Технический Университет»***

***Кафедра ИИТ***

**Лабораторная работа №15**

**По дисциплине ОАиП за II семестр**

**Тема: «Динамические структуры. Деревья»**

**Выполнил:**

Студент 1-го курса

Группы ИИ-15(1)

Волк И. А.

**Проверила:**

Хацкевич М. В.

Брест 2018

Цель: научиться работать с деревьями.

27. Напишите программу, содержащую процедуру Create(T,n), где n – положительное целое число, которая строит Т – дерево, изображенное на рисунке. В программе используйте подпрограммы.



Код программы:

#include "stdafx.h"

#include <Windows.h>

#include <iostream>

#include <cmath>

using namespace std;

struct T

{

int content;

T \* left = NULL;

T \* right = NULL;

};

void Create(T\*, int);

void createTree(T\*, int, int);

void addBranches(T\*, int);

void outputTree(T\*);

bool outputLayer(T\*, int, int now = 1);

int main()

{

T\* t = new T;

Create(t, 5);

outputTree(t);

system("pause");

return 0;

}

void Create(T\* t, int n)

{

createTree(t, n, 1);

}

void createTree(T\* t, int n, int iteration)

{

if (n == 1) return;

if (iteration == 1)

t->content = iteration;

addBranches(t, iteration + 1);

createTree(t->left, n - 1, iteration + 1);

createTree(t->right, n - 1, iteration + 1);

}

void addBranches(T\* t, int n)

{

t->left = new T;

t->right = new T;

t->left->content = n;

t->right->content = n;

}

void outputTree(T\* t)

{

int l = 1;

while (outputLayer(t, l++)) cout << endl;

}

bool outputLayer(T\* t, int need, int now)

{

if (need == now)

{

cout << t->content << " ";

return true;

}

if (need != now&&t->left == NULL&&t->right == NULL)

return false;

bool isL = outputLayer(t->left, need, now + 1);

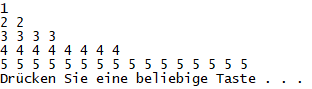
if (!isL) return false;

outputLayer(t->right, need, now + 1);

return true;

}

Результат:



Вывод: научился работать с деревьями.

bool outputLayer(T\* t, int need, int now):

