

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных средств

## **ОТЧЕТ**

по лабораторной работе №8

«Бинарные деревья»

Выполнил  
студ. гр.150702 (пг1)  
Ефимчик А.И.

Проверил  
ст. преп. каф. ЭВС  
Демидович Г.Н.

Минск 2022

## 1 ЦЕЛЬ И ЗАДАЧИ ЛАБОРАТОРНОЙ РАБОТЫ

**1.1** Целью лабораторной работы является изучение бинарных деревьев и формирование практических навыков разработки алгоритмов и компьютерных программ с использованием бинарных деревьев.

**1.2** Для достижения поставленной цели необходимо решить следующие задачи:

1) Дополнить и расширить сведения по теме ЛР из учебного пособия [1].

**1.3** Выполнить следующие задания по ЛР в соответствии с вариантом №5, разработав алгоритмы их реализации, запрограммировав их с использованием языка «Си», отладив и представив результаты работы компьютерных программ.

**Задание 1** Написать функцию для добавления элементов в дерево. Написать функцию для удаления всех элементов дерева(должна выполняться перед выходом из программы). Вывести дерево на экран.

## 2 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

### 2.1 Результат выполнения задания по заполнению и выводу на экран бинарного дерева

**2.1.1** На рисунке 1 приведена блок-схема алгоритма для выполнения задания №1 (добавление элемента в дерево и полная очистка дерева).

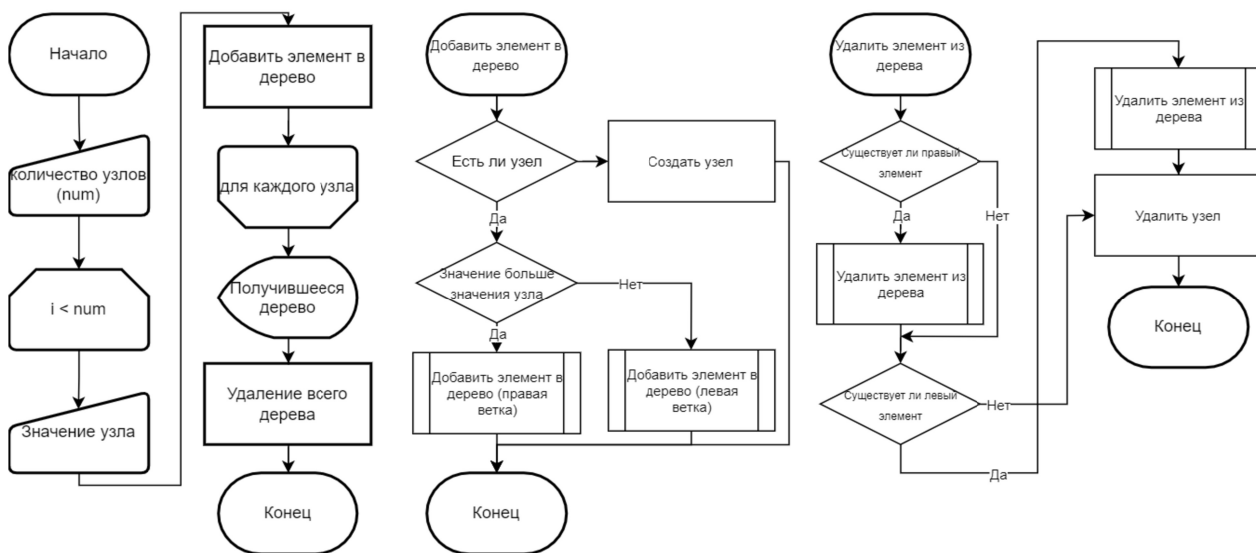


Рисунок 1 - Блок-схема алгоритма добавления элемента в дерево вывода на экран и полного удаления бинарного дерева

2.1.2 Листинг компьютерной программы по заданию 1 (добавление, удаление и вывод элементов дерева) .

```
struct Node
{
    int value;
    struct Node *Left;
    struct Node *Right;
};

void addTreeElement(struct Node **TreeRoot, int value);
void deleteTreeElement(struct Node *TreeRoot);
void printTree( struct Node* treePtr, int spaces );

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "Strings.h"
#include "Tree.h"

int main()
{
    printf("LAB 8.1 by Efimchik Alexandr from GROUP 150702 \n");
    printf("Enter nodes num: ");
    int num;
    scanf("%d", &num);

    struct Node *Root = NULL;

    for (int i = 0; i < num; i++)
    {
        printf("Enter %d node value: \n", i + 1);
        int n;
        scanf("%d", &n);
        addTreeElement(&Root, n);
    }
    printf("\nPrinted Tree:\n");
    printTree(Root, 5);
    deleteTreeElement(Root);

    return 0;
}

#include "Tree.h"
```

```

#include <stdio.h>
#include <stdlib.h>

void addTreeElement(struct Node **TreeRoot, int value)
{
    if (!(*TreeRoot))
    {
        struct Node *NewNode = calloc(1, sizeof(struct Node));
        NewNode->value = value;
        NewNode->Left = NULL;
        NewNode->Right = NULL;
        *TreeRoot = NewNode;
        return;
    }
    if (value > (*TreeRoot)->value)
    {
        addTreeElement(&((*TreeRoot)->Right), value);
    }
    else
    {
        addTreeElement(&((*TreeRoot)->Left), value);
    }
}

void deleteTreeElement(struct Node *TreeRoot)
{
    if (TreeRoot->Right != NULL)
    {
        deleteTreeElement(TreeRoot->Right);
    }
    if (TreeRoot->Left != NULL)
    {
        deleteTreeElement(TreeRoot->Left);
    }
    free(TreeRoot);
}

void printTree(struct Node *treePtr, int spaces)
{
    int loop; // loop counter

    // while not the end of tree
    while (treePtr != NULL)
    {

```

```

// recursive call with right subtree
printTree(treePtr->Right, spaces + 4);

// loop and output spaces
for (loop = 1; loop <= spaces; loop++)
{
    printf(" ");
} // end for
printf("%d\n", treePtr->value);

// set pointer to left subtree and make recursive call
printTree(treePtr->Left, spaces + 4);
treePtr = NULL;
} // end while
}

```

**2.2.3 Результаты выполнения компьютерной программы по заданию 1 в виде «скрин-шота» изображения на мониторе представлены на рисунке 2:**

```

LAB 2.2 by Efimchik Alexandr from GROUP 150702
Enter figures num: 3
Enter 1 figure name: f1
What would you like to enter:
1)Perimeter
2)Color
?
1
Enter 1 figure perimeter: 123.45
Enter 2 figure name: f2
What would you like to enter:
1)Perimeter
2)Color
?
2
Enter 2 figure color: sd
Invalid color (need to be hex)
ffaaaff
Entered: ffaaff
Enter 3 figure name: f3
What would you like to enter:
1)Perimeter
2)Color
?
1
Enter 3 figure perimeter: 15.4

```

```

SuperStructSoftware
1) Filter figures by perimeter
2) Delete figures by color
3) View structs
4) Exit
1
Enter filter perimeter: 50
Figures, with perimeter higher than 50.000000:
=====
| Name:   f1 | Perimeter: 123.45000 |
=====

```

```

SuperStructSoftware
1) Filter figures by perimeter
2) Delete figures by color
3) View structs
4) Exit
2
Specify delete filter: ffaaff
=====
| Name:   f1 | Perimeter: 123.45000 |
| Name:   f3 | Perimeter: 15.40000 |
=====

```

а)

б)

в)

Рисунок 2 - Скрин-шот выполнения операций: а) заполнения массива структур пользовательскими данными; б) фильтрация фигур с заполненным полем «периметр» по заданному значению; в) удаление элемента массива с фигурой заданного названия

## 2.2 Выводы по результатам выполнения ЛР

В результате выполнения ЛР изучены методы работы с бинарными деревьями (принцип работы, причины использования, плюсы и минусы этого подхода), получены практические навыки по написанию функций работы с бинарными деревьями на языке С.

### **3 КОНТРОЛЬНЫЕ ВОПРОСЫ И ПРЕДЛОЖЕНИЯ**

**3.1** Пояснить основные положения, термины и определения в материалах лекции (лекций) и литературе по теме ЛР.

**3.2** Объяснить алгоритмы выполнения заданий, указанных в данном варианте ЛР.

**3.3** Прокомментировать листинги (фрагменты листингов) компьютерных программ в данном варианте ЛР.

**3.4** Прокомментировать результаты выполнения заданий, указанных в варианте ЛР.

### **4 СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Луцик, Ю. А. Основы алгоритмизации и программирования : язык Си : учебно-метод. пособие / Ю. А. Луцик, А. М. Ковальчук, Е. А. Сасин. – Минск : БГУИР, 2015. – 170с. : ил.