Изображение выглядит как цепь

Автоматически созданное описание

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| УГНС | | | 09.00.00 | Информатика и вычислительная техника | | |
| Направление подготовки | | | 09.03.01 | Информатика и вычислительная техника | | |
| Направленность (профиль) | | |  | Автоматизированные системы обработки информации и управления | | |
| Форма обучения | | |  | очная | | |
|  | | |  |  | | |
| Факультет | | |  | Информационных технологий и управления | | |
| Кафедра | | |  | Систем автоматизированного проектирования и управления | | |
| Учебная дисциплина | |  | Разработка программных систем | | |
| Курс | I | | | Группа | 403 |

Отчёт по лабораторной работе № 1

Вариант № 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Исполнитель: |  |  |  |  |
| обучающийся группы 403 |  |  |  | Старков Силантий Денисович |
|  |  | (дата, подпись) |  |  |
|  |  |  |  |  |
| Проверили: |  |  |  | Корниенко Иван Григорьевич |
|  |  | (дата, подпись) |  | Федин Алексей Константинович |

Оглавление

[1. Постановка задачи 3](#_Toc96990129)

[2. Исходный данные 3](#_Toc96990130)

[3. Особые ситуации 3](#_Toc96990131)

[4. Математические методы и алгоритмы решения задач 3](#_Toc96990132)

[5. Форматы представления данных 4](#_Toc96990133)

[6. Структура программы 4](#_Toc96990134)

[7. Блок-схемы алгоритмов программы 6](#_Toc96990135)

[8. Описание хода выполнения лабораторной работы 8](#_Toc96990136)

[9. Результаты работы программы 8](#_Toc96990137)

[10. Исходный текст программы 10](#_Toc96990138)

## Постановка задачи

Необходимо реализовать структуру данных «Двоичное дерево поиска», продемонстрировать ее характерные особенности, реализовать возможность добавления и удаления элементов, визуализировать дерево. В программе должны быть предусмотрены три варианта заполнения: пользователем с клавиатуры, из файла и случайными числами.

## Исходный данные

В качестве исходных данных программа использует вводимую пользователем последовательность чисел, в программе она парсится и каждое числовое значение, содержащееся в ней, становится частью объекта класса Node и представляет значение узла в дереве. В этом классе также хранится информация о ближайших большем и меньшем узле, а также информация о родителе узла, если таковые имеются.

## Особые ситуации

Необходимо рассмотреть следующие особые ситуации.

* Если пользователь ввёл значение, не соответствующее операбельному типу, вычислений не происходит, программа просит ввести корректное значение, либо выполняет преобразование к нужному типу.
* Если пользователь ввёл несколько одинаковых значений, в дерево попадает только первая итерация данного числа.
* Если при файловом вводе или при удалении или добавлении элемента возникает конфликт построения дерева, программа выводит сообщение об ошибке и переносит пользователя в главное меню.

## Математические методы и алгоритмы решения задач

Согласно постановке задачи для составления программы будет использовано двоичное дерево поиска.

У каждого узла будут члены, содержащие ссылки на правый элемент, левый элемент и элемент-родителя при их наличии.

Добавление, поиск и удаление узлов будут осуществляться с помощью рекурсии и проверки на соответствие числовых значений узлов.

:

## Форматы представления данных

Программа использует следующие переменные:

Таблица 1 – Переменные, используемые в программе

|  |  |  |
| --- | --- | --- |
| Имя | Тип | Описание |
| Right | Node | Член-данное класса Node, хранящий правый узел от данного |
| Left | Node | Член-данное класса Node, хранящий левый узел от данного |
| Parent | Node | Член-данное класса Node, хранящий ссылку на родителя этого узла |
| Data | int | Член-данное класса Node, хранящий собственное значение |
| tree | BinaryTree | Объект, представляющий методы по добавлению, поиску и удалению элемента классу Node |
| root | Node | Объект – член класса BinaryTree, являющийся корнем всего дерева. |
| pictureBox | PictureBox | Контейнер, необходимый для визуализации дерева |
| NodeDraw | NodeForDraw | Объект, представляющий собой узел с собственным цветом, положением |
| nodeListLines | List  <List  <NodeForDraw  >> | Список линий, соединяющих узлы для отрисовки дерева |

Для хранения узлов дерева используется класс Node, в котором хранятся характеристики узла, а именно числовое значение узла, ссылки на узлы родителя, левого и правого ребенка.

Для реализации разных методов ввода используется абстрактный класс Input, от которого наследуются классы, в которых определены методы ввода с клавиатуры, из файла и случайными значениями.

## Структура программы

Программа разделена на следующие модули:

1. Program: начальный модуль с функцией Main. В нем происходит замуск главного окна приложения.
2. MainForm: в нем находятся функции, инициализирующие компоненты формы и осуществляющие связь компонентов формы с бизнес-логикой проиложения.
3. KeyboardInput, RandomInput, FileInput: модули, в которых определены методы для создания объектов класса в соответствии с выбором пользователя.
4. Tree: модуль, хранящий дерево и предоставляющий функции для работы с ним.
5. BinaryTreeTests: модуль, хранящий тесты основных расчётных функций.
6. MessageForm: модуль, инициализирующий компоненты окна с дополнительной информацией.
7. App.config: модуль, необходимый для сохранения пользовательских настроек.

Программа разделена на несколько функций:

Таблица 3 – Функции, составляющие программу

|  |  |
| --- | --- |
| Имя | Описание |
| Insert | Функция необходимая для корректного добавления новых узлов в дерево |
| Remove | Функция необходимая для корректного удаления узлов из дерева |
| Find | Функция для нахождения узла по числовому значению |
| Input | Перегруженный метод, осуществляющий генерацию массива чисел для добавления их в дерево. |
| PrintBinaryTree | Функция для корректного сохранения дерева файл |
| ShowGreeting | Функция, показывающая приветствие при запуске программы |
| SavelData | Функция, сохраняющая исходные данные |
| SavelResults | Функция, сохраняющая дерево в файл |
| PlantTree | Функция, осуществляющая отображение дерева в пользовательском интерфейсе |
| DrawLine | Функция, осуществляющая отрисовку линий между узлами в интерфейсе |
| MoveNode | Функция, позволяющая осуществлять передвижение узла пользователем |

## Блок-схемы алгоритмов программы

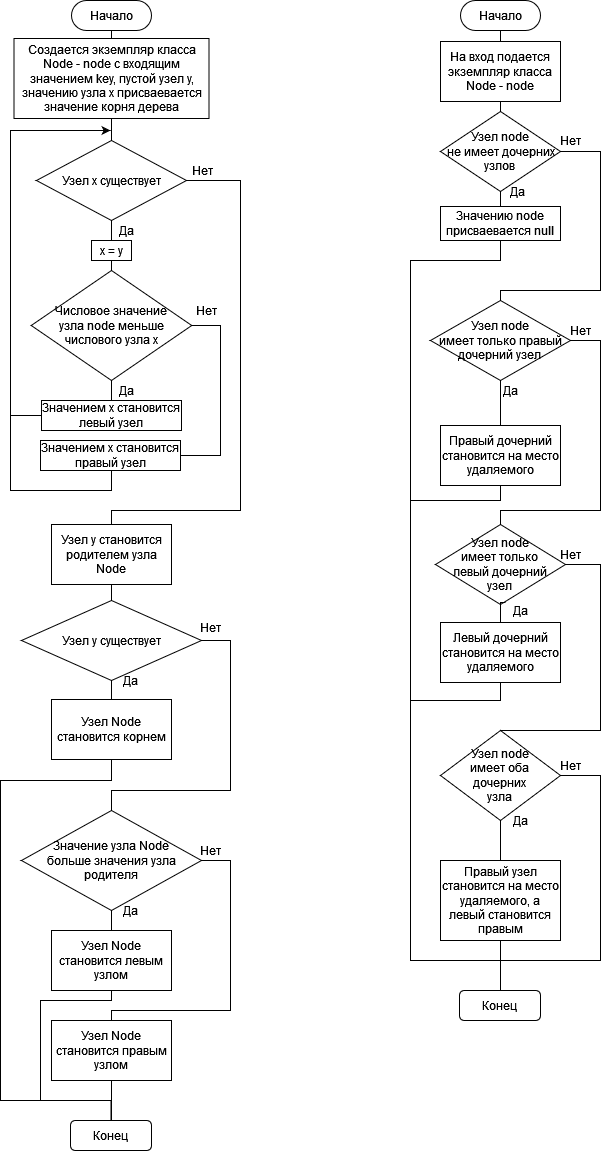


Рисунок 1 – Блок схема алгоритма добавления узла в дерево

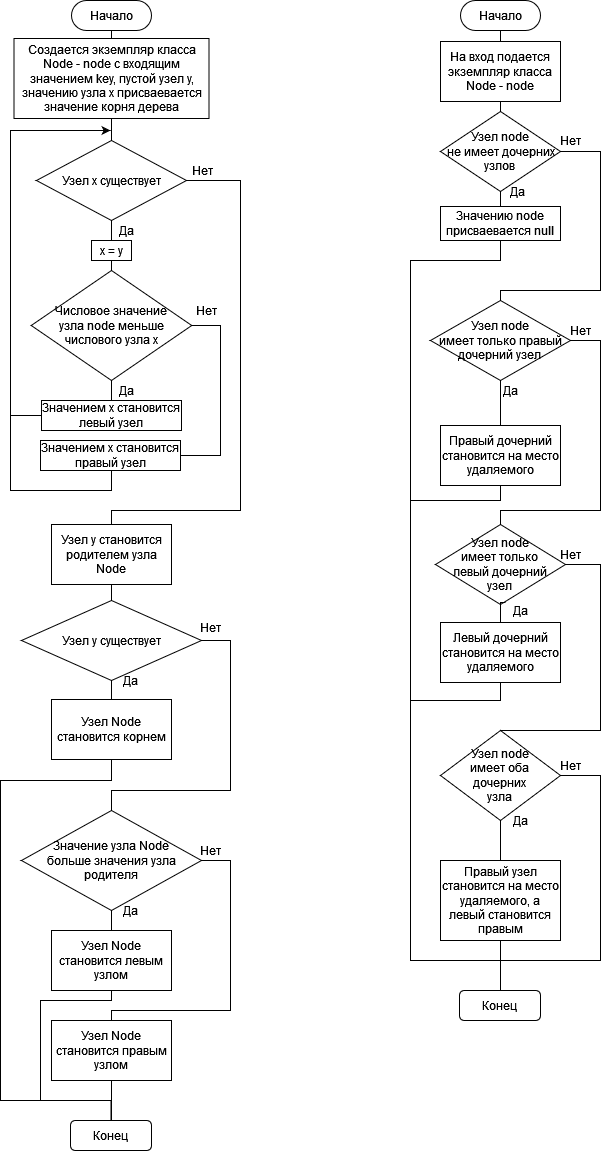


Рисунок 2 – Блок схема алгоритма удаления узла из дерева

## Описание хода выполнения лабораторной работы

1. В ходе лабораторной работы было создано решение (Solution) в интегрированной среде разработки Microsoft Visual Studio 2019. В нём был создан проект.
2. После написания приложения, было выяснено, что при движении объекта в PictureBox, происходит раздвоение этого объекта. Проблема была решена привязкой движения узла к движению курсора.
3. Была добавлена функция ClearPictureBox()для перерисовки дерева и отдельных его веток.

## Результаты работы программы

В результате работы программы, на экран выводится бинарное дерево чисел.

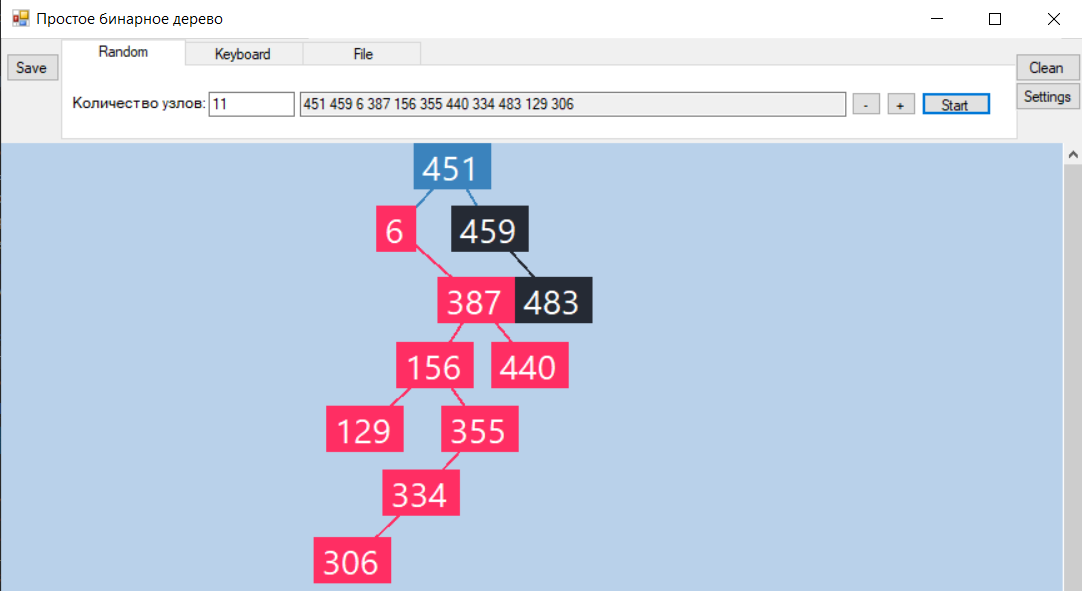


Рисунок 3 – Результат работы программы

Во всех текстовых полях осуществляется валидация входных значений, что исключает возможность добавления некорректных значений в дерево.

При добавлении в дерево уже существующего значения, появляется сообщение об ощибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 4 – Добавление элемента

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Сообщение об ошибке при добавлении уже существующего элемента

Аналогично при удалении несуществующего элемента.

При некорректности данных в открываемом файле появляется сообщение об ошибке.

**Изображение выглядит как текст

Автоматически созданное описание**

Рисунок 6 – Сообщение о некорректности вводимых данных

## Исходный текст программы