Министерство образования и науки Российской Федерации

Санкт-Петербургский политехнический университет Петра Великого

—

Институт прикладной математики и механики

**Кафедра «Информационная безопасность компьютерных систем»**

**КУРСОВАЯ РАБОТА**

По дисциплине «Структуры данных»

Выполнил

студент гр. 13656/2 <подпись> Д.В. Михайлов

Руководитель,

асс. преподавателя <подпись> А.В. Мясников

Санкт-Петербург

2018

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| Введение…………………………………………………………………. | 2 |
| 1. Термины и определения……………………………………………… | 3 |
| 2. Алгоритмы…………………………………………………………….. | 4 |
| 3. Структуры и типы данных…………………………………………… | 8 |
| 4. Описание реализации………………………………………………..... | 8 |
| 5. Заключение…………………………………………………………….. | 9 |
| 6. Список используемой литературы…………………………………… | 10 |

**ВВедение**

В качестве курсовой работы я, с разрешения преподавателя, решил реализовать утилиту “Графическое представление АВЛ-деревьев” на языке программирования «C++». Я использовал подключаемую библиотеку «Windows Forms», в которой можно создавать графический интерфейс.

**1 Термины и определения**

Перечень актуальных и необходимых для понимания концепции утилиты терминов:

1. Утилита — представленная в объективной форме совокупность данных и команд, предназначенных для функционирования ЭВМ и других компьютерных устройств с целью получения определённого результата, включая подготовительные материалы, полученные в ходе разработки программы для ЭВМ.
2. АВЛ-дерево — сбалансированное дерево, у которого для каждой вершины выполняется требование: число вершин в левом и правом поддеревьях различается не более, чем на 1.
3. Текстура — графическая картинка, загружаемая в программу.
4. Спрайт — определенная область, ограниченного размера, в которой и существует текстура.
5. Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки.

**2 Алгоритмы**

Основная составляющая реализации утилиты в библиотеке «Windows Forms» заключается в создании окон и объектов для взаимодействия пользователя с интерфейсом.

Процесс создания интерфейса утилиты осуществляется с помощью Панели инструментов, где с помощью форм Windows Forms добавляем различные формы для взаимодействия с программой (Рис. 1).

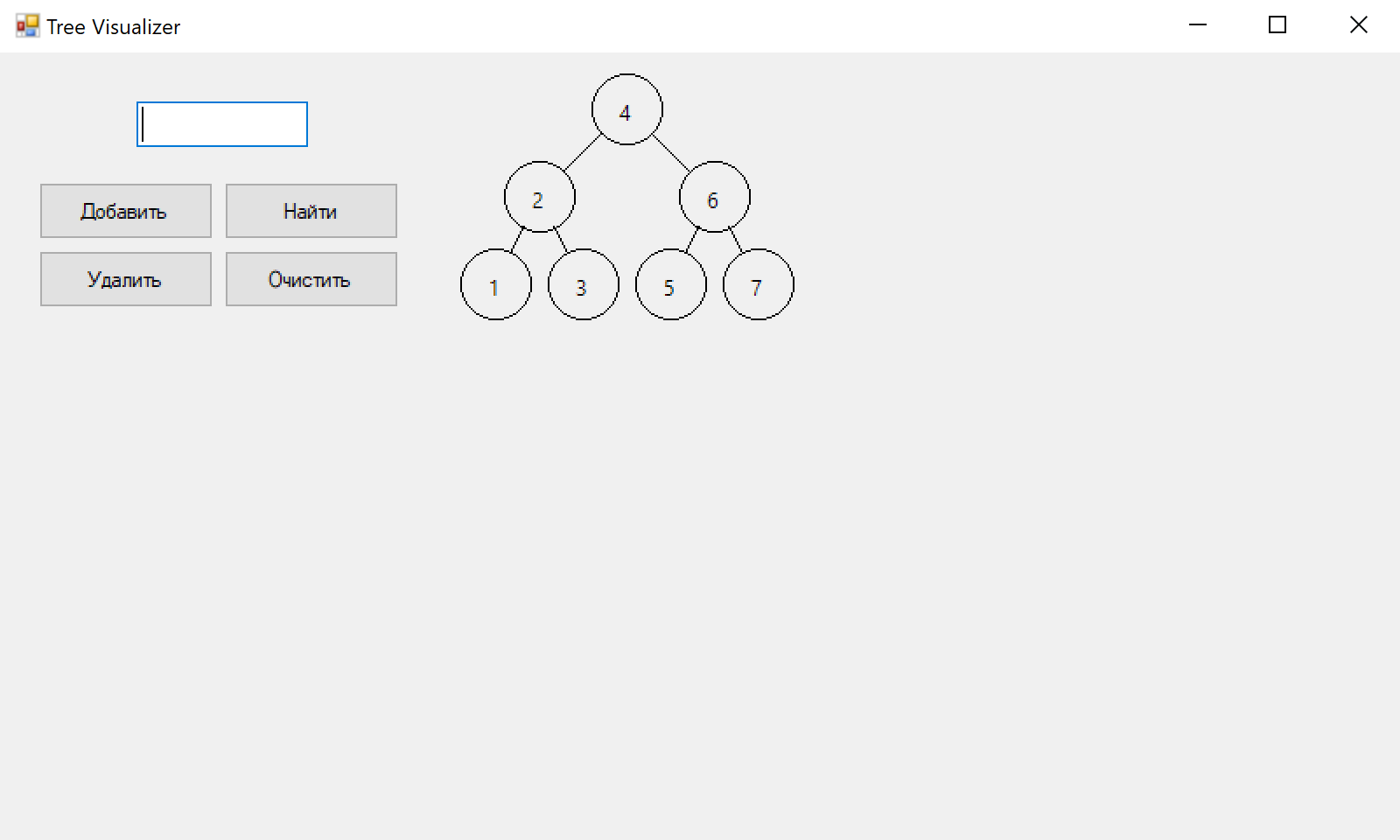
****

Рисунок 1. Главное окно утилиты

Списочное представление бинарных АВЛ-деревьев основано на элементах, соответствующих узлам дерева. Каждый̆ элемент имеет поле данных и два поля указателей̆. Один указатель используется для связывания элемента с правым потомком, а другоӗ – с левым.



Рисунок 2. Структура главных элементов дерева

Основными операциями над бинарным деревом поиска могут быть: добавление, удаление и поиск.

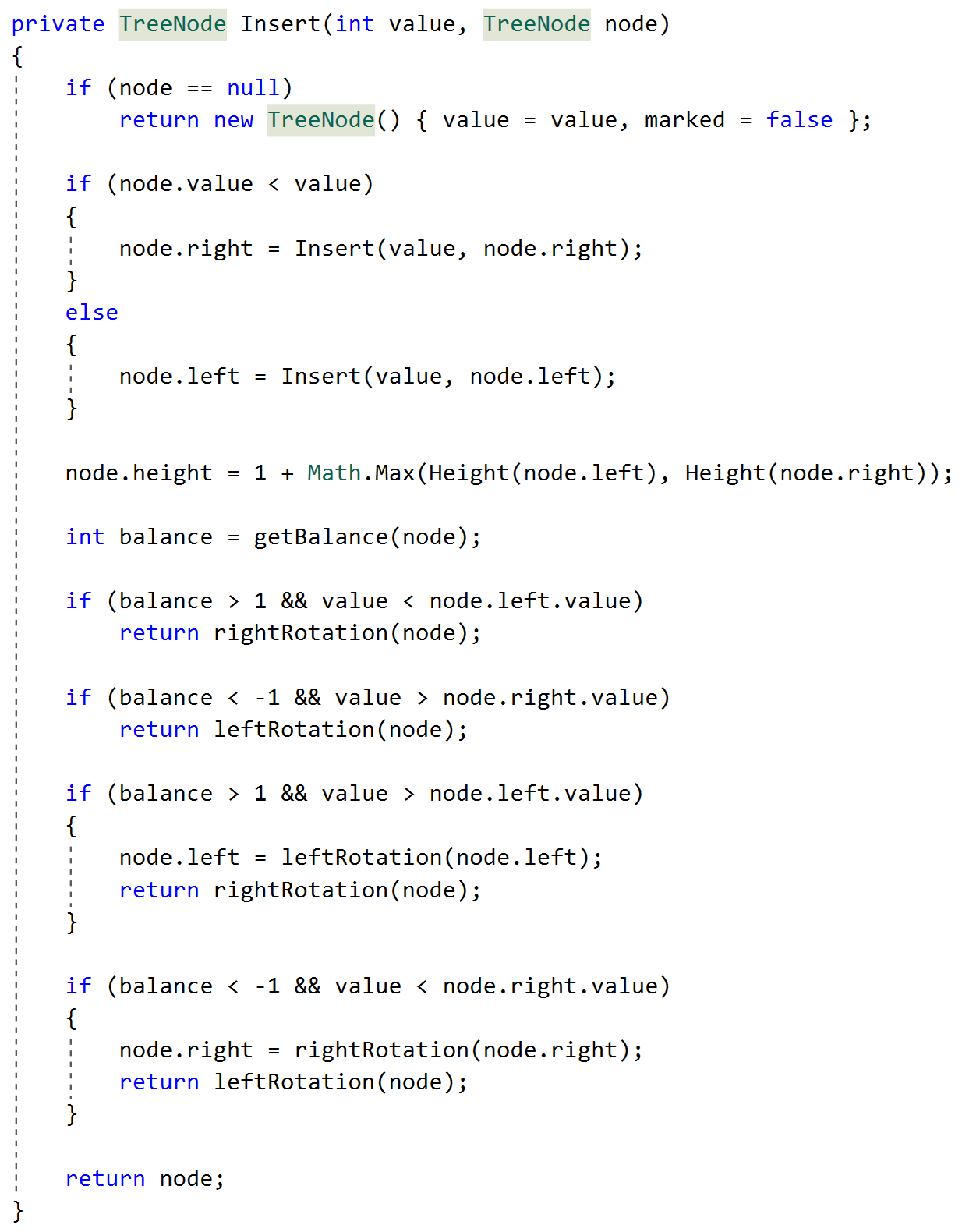


Рисунок 3. Функция добавления элемента в список “Insert”

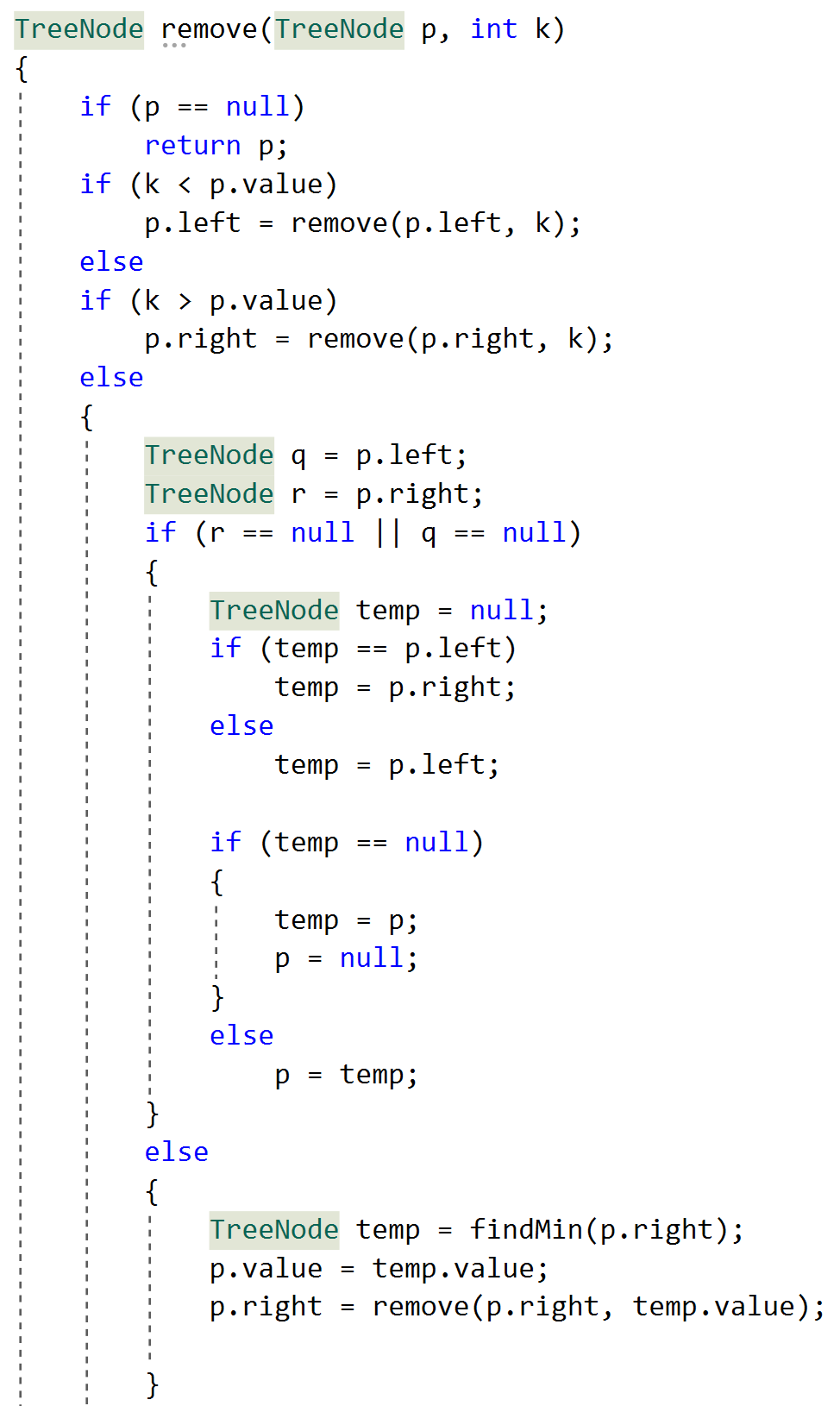


Рисунок 4. Функция удаления элемента из списка “remove”

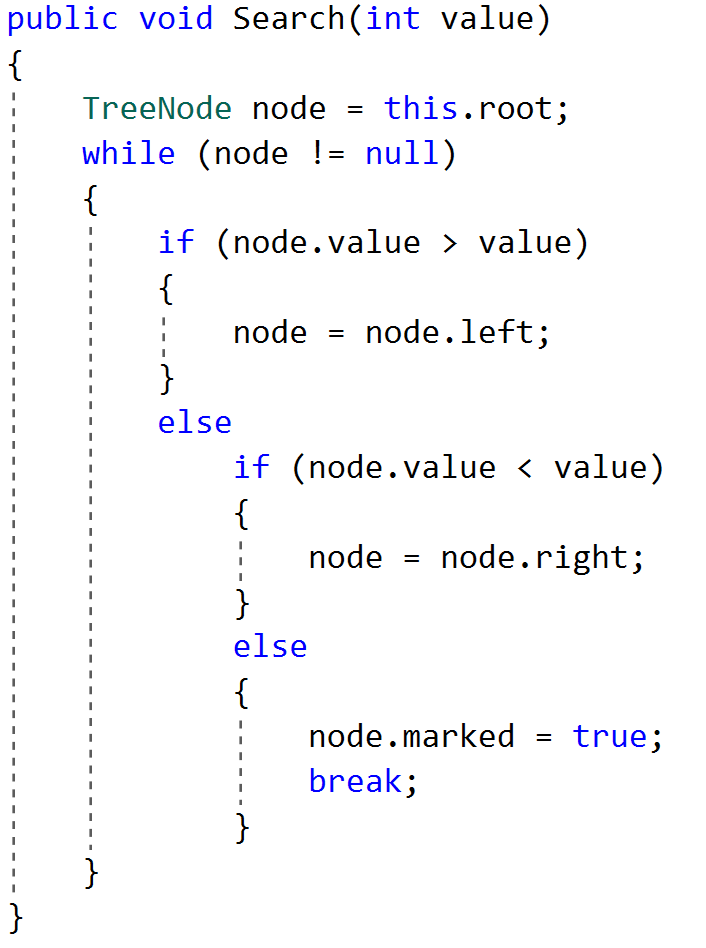


Рисунок 5. Функция поиска элемента в списке “Search”

Основным свойством АВЛ-дерева является его сбалансированность, которое реализовывается с помощью алгоритма его обхода.

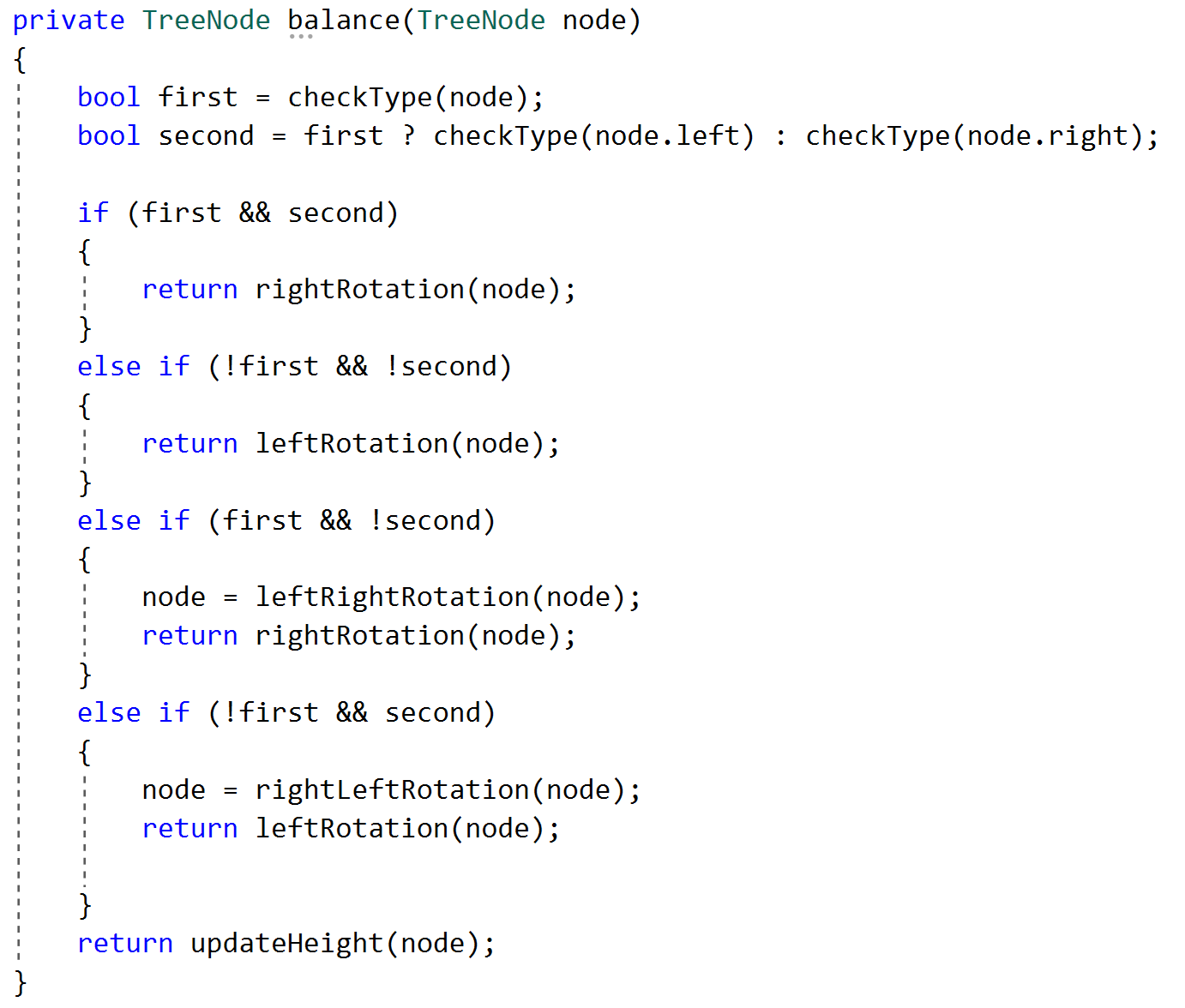


Рисунок 6. Функция балансировки дерева “balance”

**3 Структуры и типы данных**

В данной программе я использовал следующие типы данных: enum, struct, union. Каждый тип был применен максимально эффективно для реализации поставленной задачи.

Enum используется для создания информационной справки, которая поможет пользователю разобраться в главном меню. Я использовал именно enum, потому что нумерация противников это удобный способ реализации их алгоритмов в функциях.

Struct используется для создания линейных списков, в которых содержаться записи о результатах пользователей или по-другому «Таблицы лидеров».

Union используется для прямой конвертации очков пользователя, которые заданы изначально типом данных int.

Я использую логическую структуру данных «Линейный список». Она реализована с помощью встроенного контейнера «vector» и структуры struct. «vector» позволяет сортировать элементы по первому параметру в нем, а это особенно хорошо для «таблицы рекордов».

**4 Описание реализации**

В реализованной утилите мы получим окно с элементами возможной операций над деревом и окно самого рисунка дерева. У нас имеется 4 кнопки с операциями и окно ввода значений.

**5 Заключение**

Структура данных — это программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных данных в вычислительной машине. Для добавления, поиска, изменения и удаления данных структура данных предоставляет некоторый набор функций, составляющих её интерфейс.

Различные виды структур данных подходят для различных приложений; некоторые из них имеют узкую специализацию для определенных задач. Например, бинарные деревья обычно подходят для создания баз данных. Помимо этого, деревья также обычно используются при управлении иерархией данных, синтаксическом разборе арифметических выражений.

В данной работе были рассмотрены основные определения и понятия теории деревьев, а также их виды, структура АВЛ-дерева. Был рассмотрен процесс реализации графического представления дерева.

В результате выполнения работы я приобрел навыки в создании графического интерфейса, используя библиотеку Windows Forms. Научился использовать типы данных enum и union, и в целом впервые создал утилиту с графическим интерфейсом.

В ходе выполнения курсовой работы были углублены знания в разработке программ на языке программирования С++, закреплены навыки работы с файлами, изучено множество функций библиотеке Windows Forms.

**СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. P.J. Plauger. C library function - fopen(): [Электронный документ] // The Standard CLibrary (https://www.tutorialspoint.com/c\_standard\_library/ c\_function\_fopen.htm). Проверено 25.05.2018 2. Brian W. Kernigan, Dennis M. Ritchie. C - File I/O: [Электронный документ] //CProgramming Language, 2nd Edition (https://www.tutorialspoint.com/cprogramming/c\_file\_io.htm).Проверено25.05.2018

2. Jan Bodnar. Windows API dialogs: [Электронный документ] (http:// zetcode.com/gui/winapi/dialogs/). Проверено 25.05.2018

3. Microsoft Developer Network. LoadString function: [Электронный документ] (https://msdn.microsoft.com/en-us/library/windows/desktop/ms647486(v=vs. 85).aspx). Проверено 25.05.2018

3. Microsoft Developer Network. LoadBitmap function: [Электронный документ] (https://msdn.microsoft.com/en-us/library/windows/desktop/dd145033(v=vs. 85).aspx). Проверено 25.05.2018

5. Jan Bodnar. Windows API controls I: [Электронный документ] (http:// zetcode.com/gui/winapi/controls/). Проверено 25.05.2018 10. Алексей Соловьёв. Основы программирования для Win32 API: [Электронный документ] (http://dims.karelia.ru/win32/). Проверено 25.05.2018