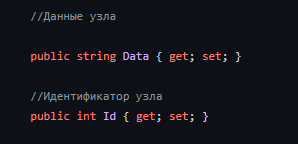
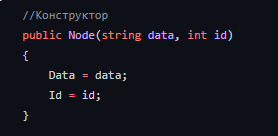
**Класс Node.cs**

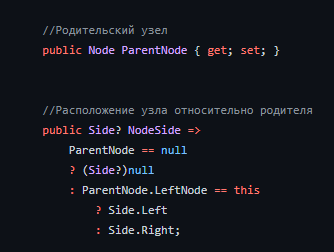
Этот класс является узлом ассоциативного контейнера, реализованного в виде дерева. Он предназначен для хранения пар ключ-значения.



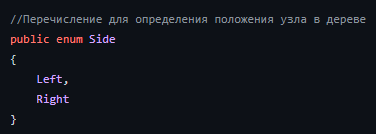
Поле Data предназначено для хранения значений контейнера, а поле Id-ключей, именно с помощью него в классе Tree реализуется поиск.



Данный метод используется в качестве конструктора класса Node, с помощью него происходит вставка данных в узел контейнера, в виде пары ключ-значение.



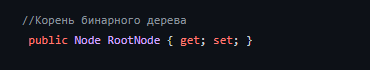
Также этот класс имеет ссылку на родительский Node, который находится наверху ассоциативного контейнера, реализованного в виде двоичного дерева, по другому его называют «корневым элементом». Метод, расположенный ниже позволяет удобно определять расположение узла относительно родительского, чтобы было понятно в какую часть дерева (правую или левую) необходимо добавить данный элемент.



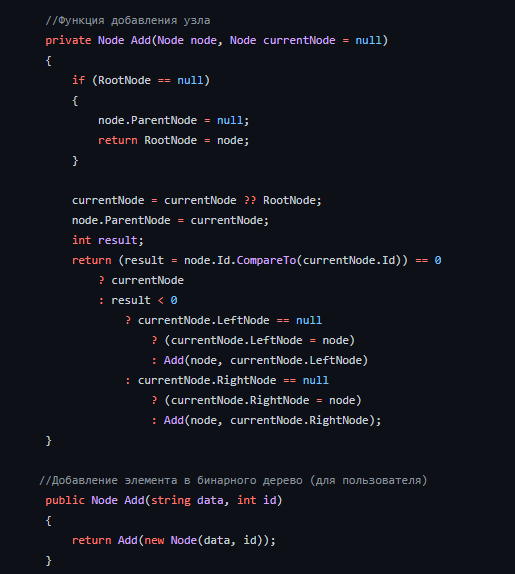
Перечисление используется для удобства обозначения ссылки на правый или левый элемент.

**Класс Tree.cs**

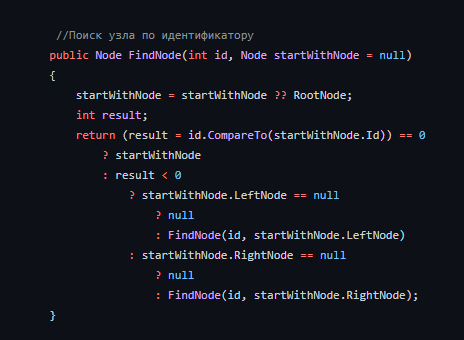
Данный класс представляет собой ассоциативный контейнер, хранящий пары ключ-значение в виде бинарного дерева, в нем реализованы необходимые методы, позволяющие осуществлять добавление, удаление и поиск элементов по ключу.



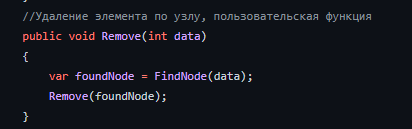
RootNode является корневым узлом, позволяющем в дальнейших методах начинать отсчет и проходиться по веткам дерева.



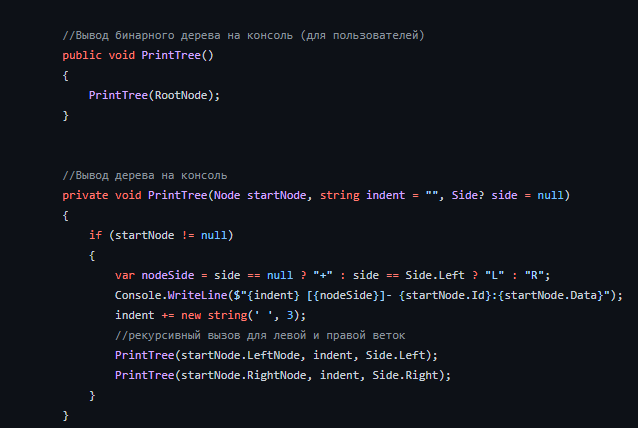
Закрытый метод Add принимает на вход какой-либо узел, и начинает обход для того, чтобы добавить элемент в дерево, публичный Add является оболочкой над закрытым методом и позволяет пользователю удобно добавить нужный элемент в ассоциативный контейнер.



Метод FindNode необходим для поиска узла по значению, он осуществляет обход дерева, начиная с корневого элемента и возвращает найденный узел, в случае его существования в рамках хранимой структуры данных.



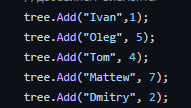
RemoveNode осуществляет удаление элемента по индексу, в данном методе сначала реализуется поиск значения по ключу, в случае успешного выполнения вызывается закрытый метод remove, который удаляет найденный элемент из ассоциативного контейнера, меняя в соседних элементах ссылки для сохранения конфигурации структуры данных.



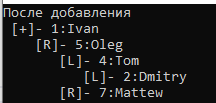
Метод PintTree необходим для вывода элементов ассоциативного контейнера в консоль, он осуществляет обход элементов, начиная с корневого узла и выводит их в соответствии с древовидной структурой.

**Класс Program.cs**

Класс Program.cs необходим для отладки и запуска программы. В приведенном примере осуществляется попытка добавления в ассоциативный контейнер 5 элементов.

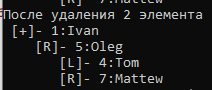


После чего конфигурация дерева выводится на экран с помощью метода PrintTree().



Как видно из рисунка, ассоциативный контейнер приобрел древовидную структуру, добавив необходимые элементы либо в левую либо в правую часть дерева.

Далее производится удаление элемента с ключом 2, в результате чего конфигурация дерева менятся с учетом исключения второго элемента, вывод так же формируется по средством использования метода PrintTree().



И последним шагом является проверка работоспособности поиска значения по индексу. Реализуем этот тест для четвертого элемента. Для этого воспользуемся методом FindNode() и выведем полученные в результате поиска данные, использовав геттер поля Data полученного Node.



Как видно из рисунка, метод поиска так же выдал верный результат, что свидетельствует о правильности работы программы.