МИНОБРНАУКИ РОССИИ

Федеральное государствен

ное бюджетное образовательное учреждение

высшего образования

**«Челябинский государственный университет»**

**(ФГБОУ ВО «ЧелГУ»)**

Институт информационных технологий

Кафедра информационных технологий и экономической информатики

ОТЧЕТ

по лабораторной работе №3

Авторы отчета С.М. Панов ПрИ-201

подпись инициалы, фамилия группа

А.Д. Казбеков ПрИ-201

подпись инициалы, фамилия группа

В.И. Кочетков ПрИ-201

подпись инициалы, фамилия группа

Отчет защищен \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

дата оценка

Челябинск 2024 г.

**Цель работы:** понять и реализовать динамические структуры данных, а так же реализовать алгоритмы работы со своими связными списками.

**Задание.**

I. **Стек.** Познакомиться со структурой данных стек.

II. **Очередь.** Познакомиться со структурой данных очередь.

III. **Понять где применяются структуры данных.**

IV. **Свой список**. Реализовать алгоритмы работы со своими списками.

V. **Бинарное дерево.** Обход бинарного дерева.

Стек технологий:

1. C#

**Часть 1**

**Задание 1:**

Реализовать Стек с помощью СВОЕГО связного списка. Должны

поддерживаться следующие операции работы со стеком: Push(elem),

Pop(), Top(), isEmpty(), Print(). (см. Рис. 1.1)

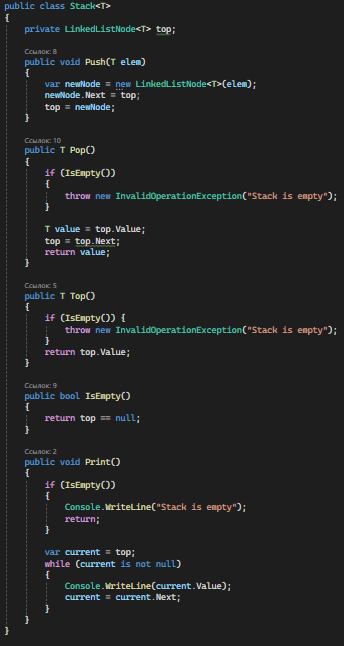


Рис. 1.1 Реализация стека

**Задание 2:**

В файле input.txt записаны числа от 1 до 5 через пробел. Каждому

числу соответствует своя операция: 1 - Push(elem), 2 - Pop(), 3 - Top(), 4

- isEmpty(), 5 - Print(). Для операции push после единицы через запятую

указывается значение помещаемого элемента (это может быть число

либо слово). Результат выполнения каждой операции выводится на

экран. (см. Рис. 2.1-2.2)

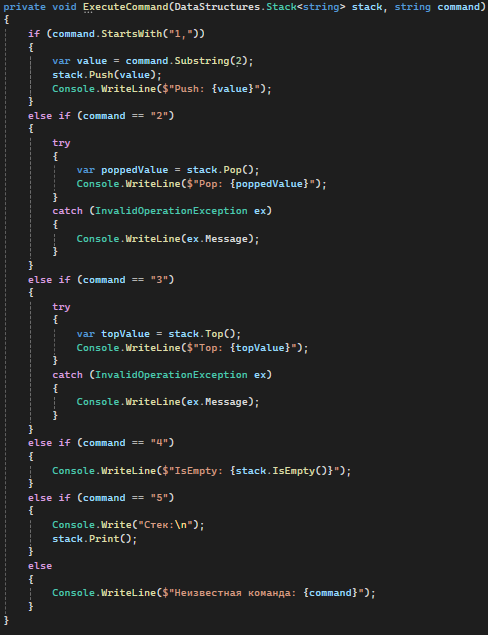


Рис. 2.1 Реализация выбора операции

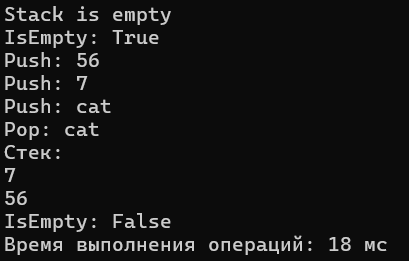


Рис. 2.2 Отображение выполнения операций.

**Задание 3:**

Произвести запуск с замером времени на считывание и выполнение

операций стеком. В отчет занести результаты замеров времени для

различных наборов операций в файле input.txt. Наборы должны быть

различными по размеру и составу операций.

Замер времени для стека. (см. Рис. 3.1)

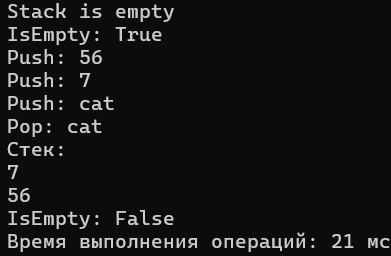


Рис. 3.1 Замер времени для стека

Замер времени для списка. (см. Рис. 3.2)

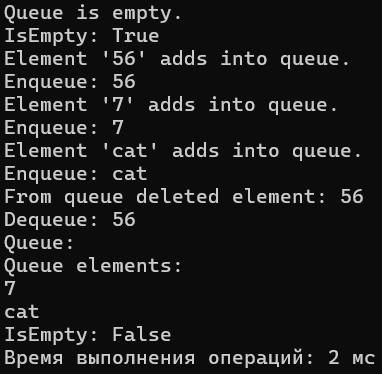


Рис. 3.2 Время выполнения для списка.

Замер времени для двусвязного списка. (см. Рис. 3.3)

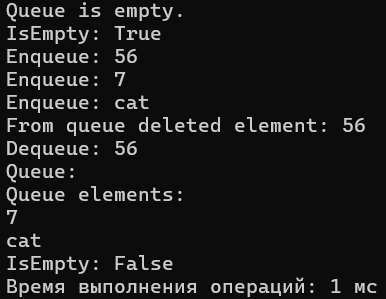


Рис. 3.3 Время выполнения для двусвязного списка.

**Задание 4.**

Реализовать алгоритм вычисления выражения, записанного в

постфиксной записи (используя постфиксные вычисления). Постфиксная запись считывается из файла. В выражение входят только

числа и знаки операций (+, -, \*, :, ^, ln, cos, sin, sqrt, «)».

После реализации алгоритма вычисления постфиксной записи,

необходимо произвести расчет оценки сложности алгоритма. (см. Рис. 4.1-4.2)

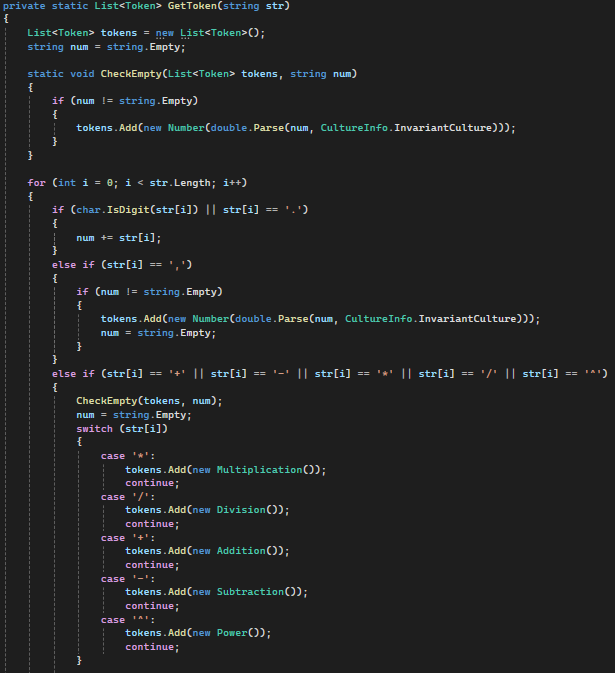


Рис. 4.1 Метод для получения токенов.

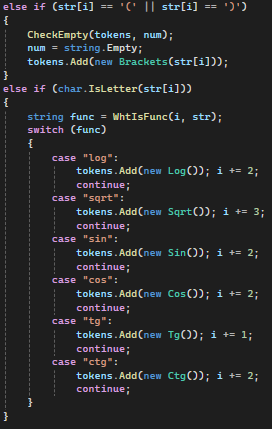


Рис. 4.2 Продолжение метода для получения токенов.

Результат выполнения. (см. Рис. 4.3)

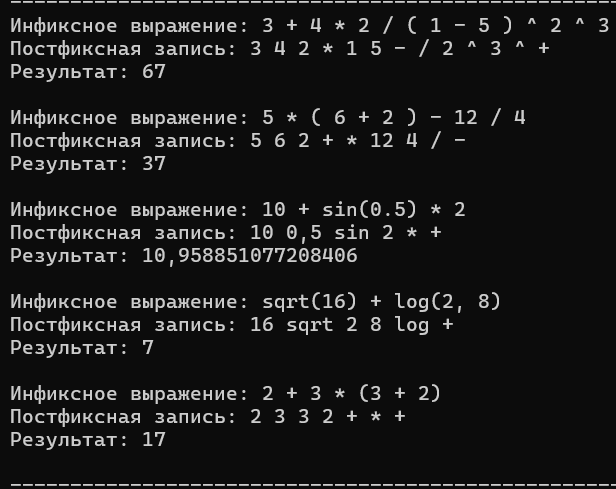


Рис. 4.3 Результат выполнения.

**Часть 2**

**Задание 1:**

Реализовать Очередь с помощью списка и с помощью стандартного

класса Queue. Должны поддерживаться следующие операции работы с

очередью: вставка/удаление элемента, проверка на пустоту, печать,

вывод первого элемента. (см. Рис. 5.1)

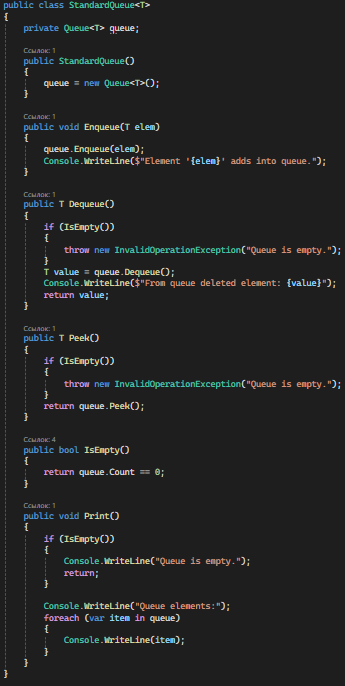


Рис. 5.1 Реализация очереди стандартным классом Queue.

Реализация очереди с помощью списка (см. Рис. 5.2)

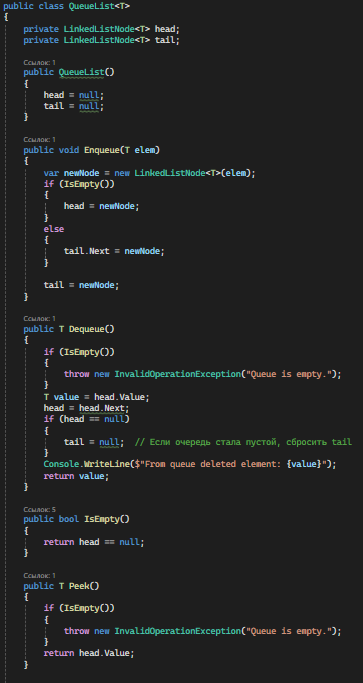


Рис. 5.2 Реализация очереди с помощью списка.

**Задание 2:**

В файле input.txt записаны числа от 1 до 5 через пробел. Каждому

числу соответствует своя операция: 1 - вставка, 2 - удаление, 3 –

просмотр начала очереди, 4 – проверка на пустоту, 5 - печать. Для

операции добавления в очередь после единицы через запятую

указывается значение помещаемого элемента (это может быть число

либо слово). Результат выполнения каждой операции выводится на

экран. (см. Рис. 5.3-5.4)

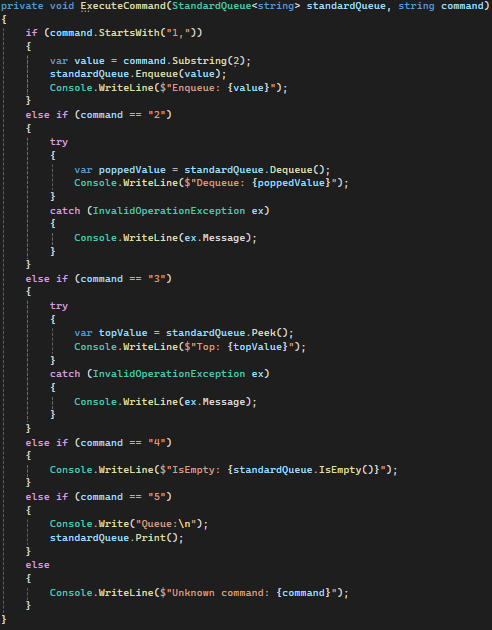


Рис. 5.3 Реализация на стандартном классе Queue

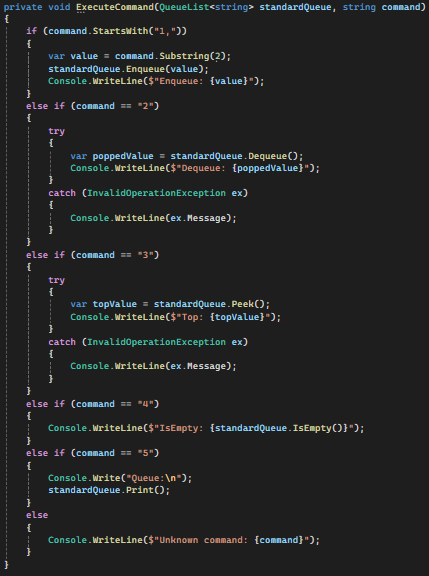


Рис. 5.4 Реализация очереди с помощью списка.

**Задание 3:**

Произвести запуск с замером времени на считывание и выполнение

операций очередью обоих реализация. В отчет занести результаты

замеров времени для различных наборов операций в файле input.txt.

Наборы операций должны быть следующими: 1) различными по длине,

2) одинаковыми по длине, но различными по составу операций (в этом

случае вам необходимо добиться максимального расхождения по

времени). Сверху изображена реализация очереди на стандартном классе Queue, снизу с помощью списка. (см. Рис. 5.5)

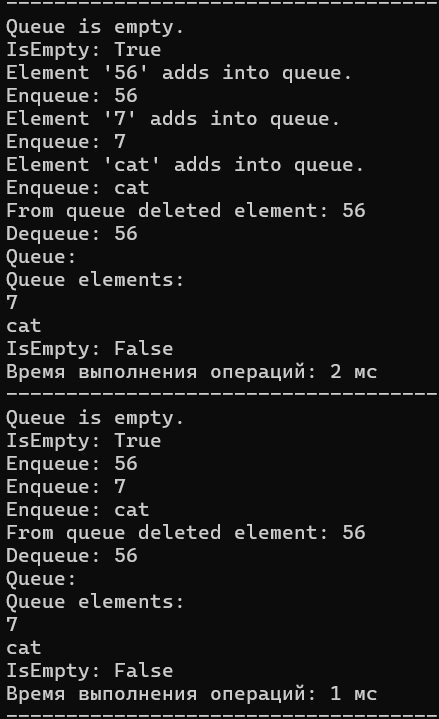


Рис. 5.5 Время выполнения.

**Часть 4.**

1. Написать функцию, которая переворачивает список L, т.е. изменяет ссылки в этом списке так, чтобы его элементы оказались расположенными в обратном порядке. (см. Рис. 6.1)

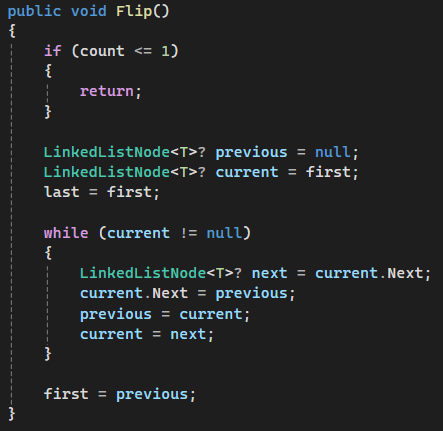


Рис. 6.1 Код функции переворота списка

1. Написать функцию, которая переносит в начало (в конец) непустого списка L его последний (первый) элемент. (см. Рис. 6.2.1 и Рис. 6.2.2)



Рис. 6.2.1 Код функции переноса последнего элемента на первый

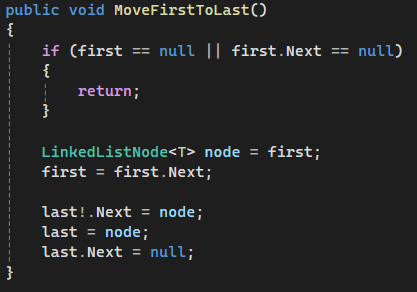


Рис. 6.2.2 Код функции переноса первого элемента на последний

1. Написать функцию, которая определяет количество различных элементов списка, содержащего целые числа. (см. Рис. 6.3)

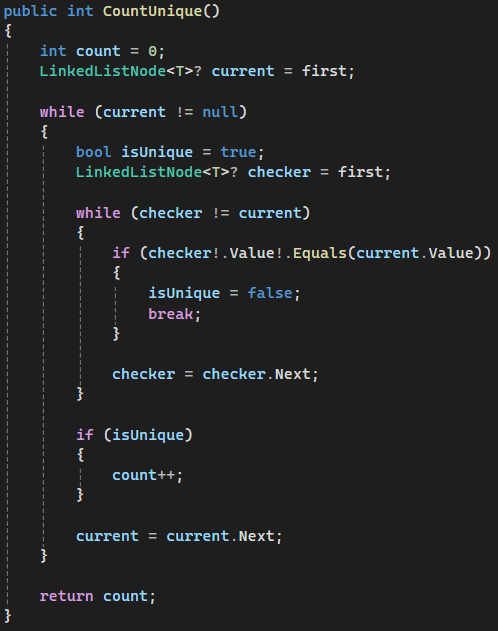


Рис. 6.3 Код подсчёта уникальных элементов

1. Написать функцию, которая удаляет из списка L неуникальные элементы.

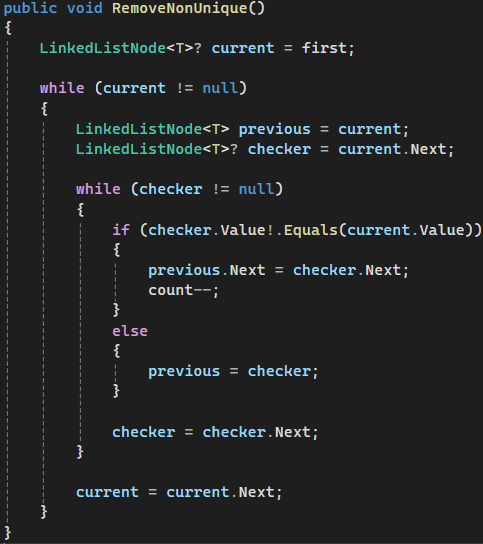


Рис. 6.4 Код удаления не уникальных элементов

1. Написать функцию вставки списка самого в себя вслед за первым вхождением числа. (см. Рис. 6.5)

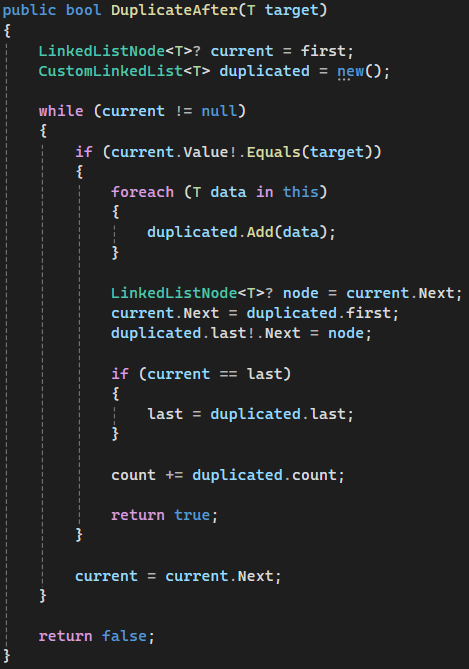


Рис. 6.5 Код вставки списка самого в себя вслед за первым вхождением числа

1. Написать функцию, которая вставляет в непустой список L, элементы которого упорядочены по не убыванию, новый элемент Е так, чтобы сохранилась упорядоченность. (см. Рис. 6.6)

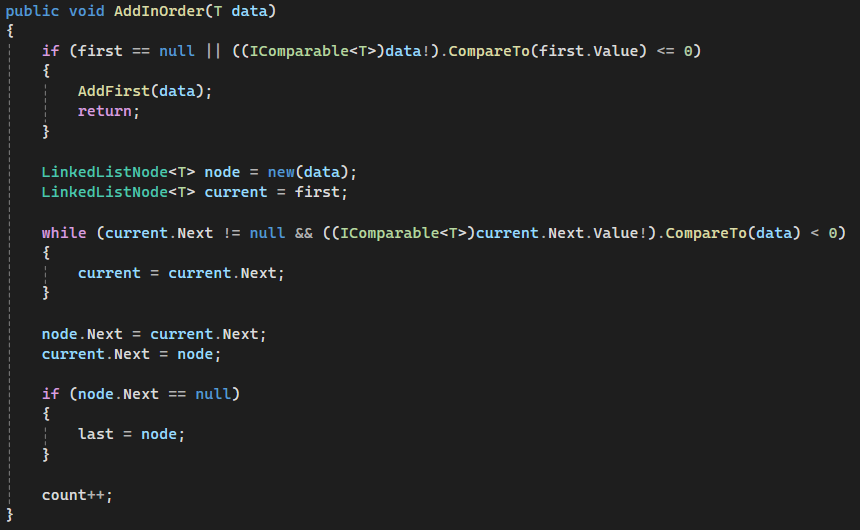


Рис. 6.6 Код вставки нового элемента с сохранением упорядоченности

1. Написать функцию, которая удаляет из списка L все элементы Е, если таковые имеются. (см. Рис. 6.7)

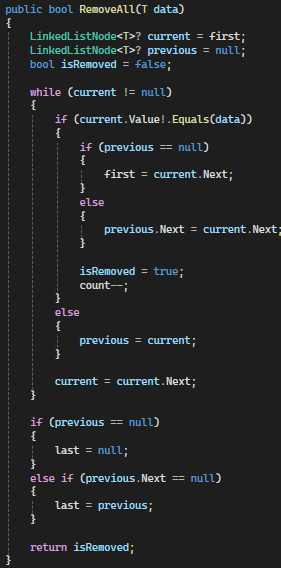


Рис. 6.7 Код удаления всех элементов Е

1. Написать функцию, которая вставляет в список L новый элемент F перед первым вхождением элемента Е, если Е входит в L. (см. Рис. 6.8)

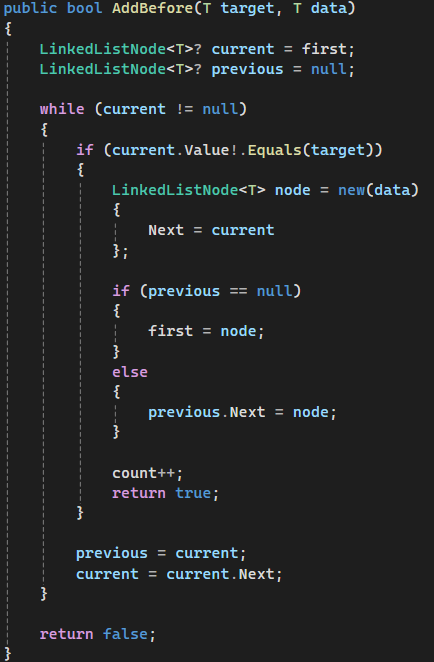


Рис. 6.8 Код вставки в список L новый элемент F перед первым вхождением элемента Е, если Е входит в L

1. Функция дописывает к списку L список E. Оба списка содержат целые числа. В основной программе считать их из файла. (см. Рис. 6.9)

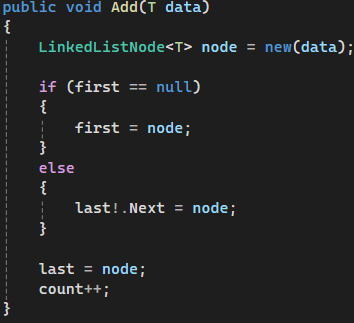


Рис. 6.9 Код дописывание к списку L список Е

1. Функция разбивает список целых чисел на два списка по первому вхождению заданного числа. Если этого числа в списке нет, второй список будет пустым, а первый не изменится. (см. Рис. 6.10)

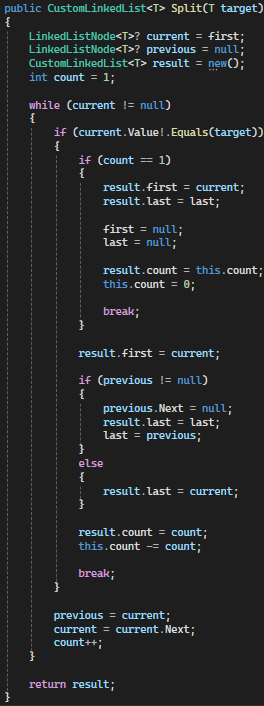


Рис. 6.10 Код разбиения списка

1. Функция удваивает список, т.е. приписывает в конец списка себя самого. (см. Рис. 6.11)

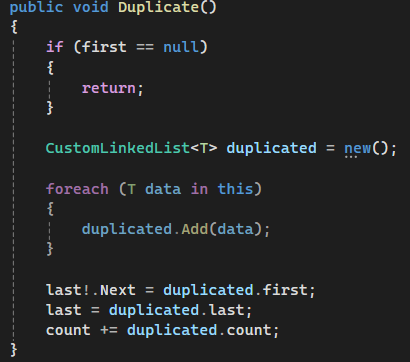


Рис. 6.11 Код удваивания списка

1. Функция меняет местами два элемента списка, заданные пользователем. (см. Рис. 6.12)

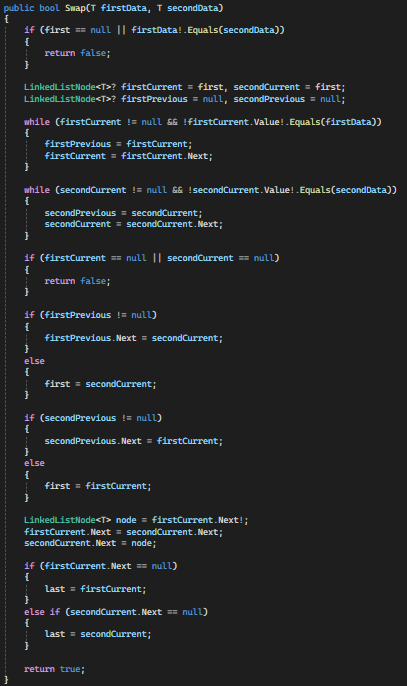


Рис. 6.12 Код замены местами два элемента списка

**Часть 5**

Задание 1 обход бинарного дерева (см. Рис. 7.1)

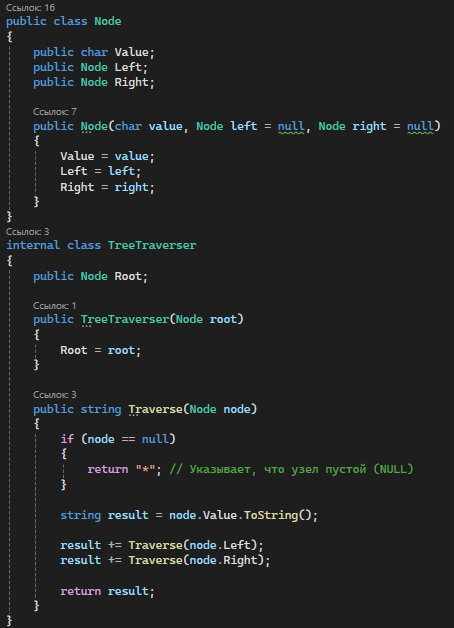


Рис. 7.1 Код обхода бинарного дерева

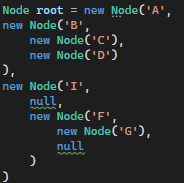


Рис. 7.2 Способ задания бинарного дерева