

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**

|  |
| --- |
|  |

Институт информационных технологий

Кафедра математического обеспечения и стандартизации

информационных технологий (МОСИТ)

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5**

«Однонаправленный динамический список»

**по дисциплине**

«Структуры и алгоритмы обработки данных»

Выполнил студент группы ИКБО-02-21 Хитров Н. С.

Принял доцент кафедры МОСИТ, к. п. н. Рысин. М. Л.

Практическая работа выполнена «\_\_» \_\_\_\_\_\_\_ 2022 г.

«Зачтено» «\_\_» \_\_\_\_\_\_\_ 2022 г.

Москва 2022

**Содержание**

[Цель работы 3](#_Toc101634628)

[Ход работы 4](#_Toc101634629)

[Определение структуры узла 4](#_Toc101634630)

[Определение операций над списком 4](#_Toc101634631)

[Алгоритм выполнения операций 7](#_Toc101634632)

[Таблица с тестированием операций 12](#_Toc101634633)

[Код программы 15](#_Toc101634634)

[Тестирование 21](#_Toc101634635)

[Вывод 25](#_Toc101634636)

Цель работы

Получить знания и практические навыки управления динамическим однонаправленным списком.

Персональныйвариант – №11.

Дан линейный однонаправленный список L, информационная часть которого содержит однозначные и двузначные числа.

1. Разработать функцию, которая создает массив А из 10 указателей на элемент списка и включает в список элемента массива с индексом i, числа списка L, которые начинаются с цифры равной i. Включение в конец списка. Однозначные числа включаются в список массива с индексом 0.

2. Разработать функцию, которая удаляет список L.

3. Разработать функцию, которая создает список L, включая в него списки массива А последовательно от списка с индексом 0 до списка с индексом 9.

Ход работы

Определение структуры узла

Информационное поле val хранит числовое значение, поле next – указатель на следующий узел (по умолчанию задано значение nullptr), Node – конструктор экземпляра узла списка.

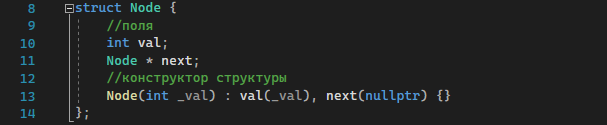


Рисунок 1 – Структура Node

Определение операций над списком

1. Вывод списка
2. Вставка нового узла перед первым узлом
3. Вставка нового узла после последнего узла
4. Поиск узла по ключу
5. Удаление узла по ключу
6. Разработать функцию, которая создает массив А из 10 указателей на элемент списка и включает в список элемента массива с индексом i, числа списка L, которые начинаются с цифры равной i. Включение в конец списка. Однозначные числа включаются в список массива с индексом 0.
7. Разработать функцию, которая удаляет список L.
8. Разработать функцию, которая создает список L, включая в него списки массива А последовательно от списка с индексом 0 до списка с индексом

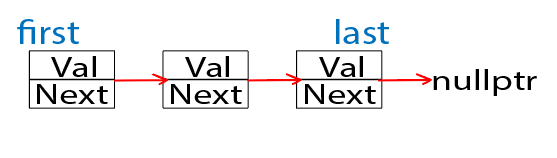


Рисунок 2 – Изображение односвязного списка

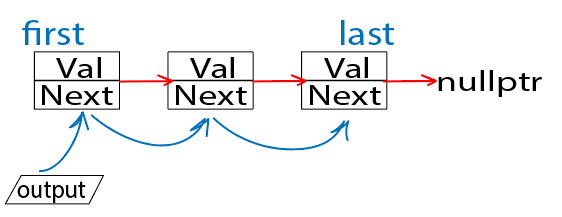


Рисунок 3 – Вывод списка

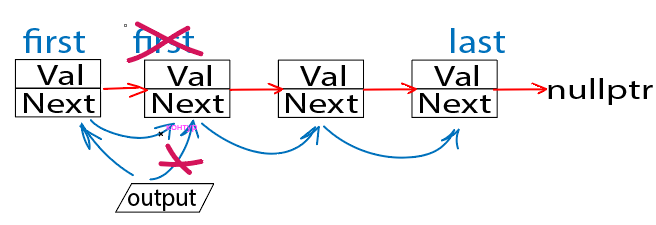


Рисунок 4 – Вставка нового узла перед первым

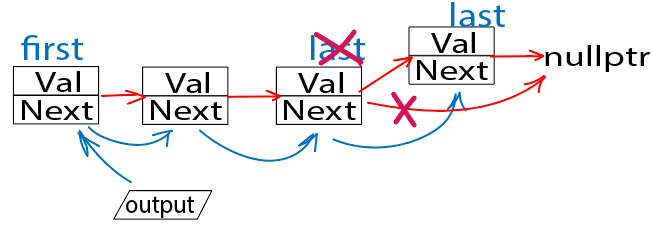


Рисунок 5 – Вставка нового узла после последнего

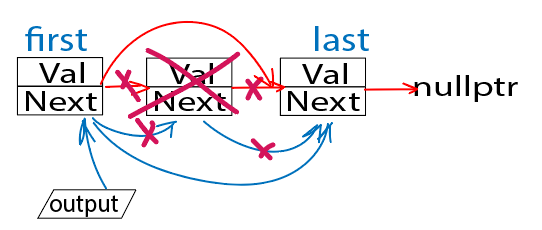


Рисунок 6 – Удаление узла по ключу

Случай: общий

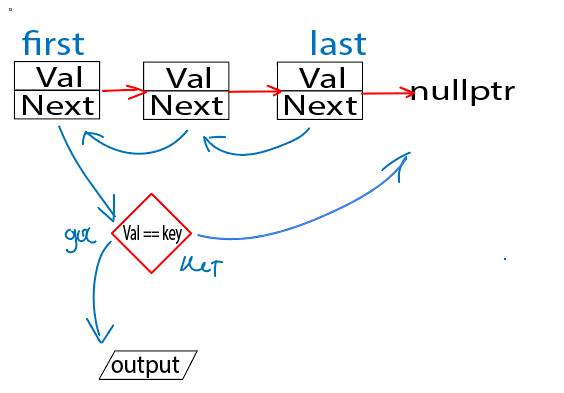
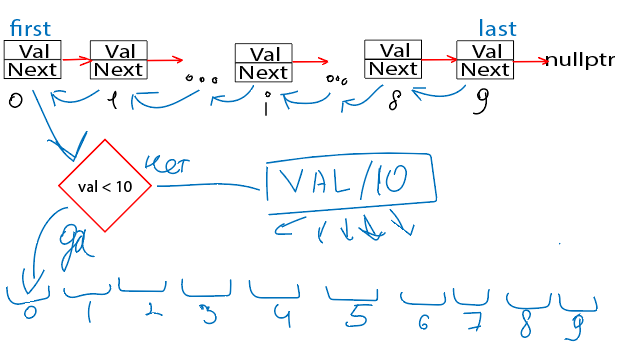


Рисунок 7 – Поиск узла по ключу



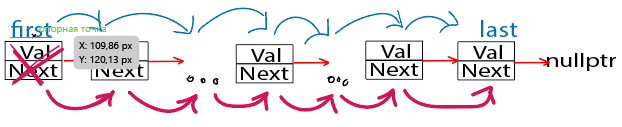
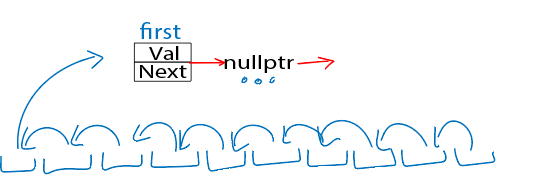


Рисунок 10 – Перенос подсписков из массива в новый список

Рисунок 9 – Удаление изначального списка

Рисунок 8 – создание массива подсписков согласно заданию 1

Таблица с тестированием операций

Таблица 10 – Тестирование функций

|  |  |  |
| --- | --- | --- |
| Функция | Входные данные | Выходные данные |
| Вставка узла в начало | 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **1.5** | **1.5** 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2 |
| Вставка узла в конец | 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **1.5** | 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2 **1.5** |
| Удаление по ключу | **2.5** 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **2.5** | 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2 |
| 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **10** | Ошибка удаления (элемент отсутствует в списке) |
| Поиск по ключу | **2.5** 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **2.5** | Элемент присутствует в списке |
| 2.5 26.22 4.6 31 40.69 91.62 44.53 9.4 80.18 7.2  **10** | Элемент отсутствует в списке |
| Создание массива списков в соответствии с заданием 1 | 72 58 57 42 28 80 53 91 88 63 51 | 28  42  58 57 53 51  63  72  80 88  91 |
| Удаление изначального списка | 72 58 57 42 28 80 53 91 88 63 51 | Все элементы списка удалены |
| Перенос списков из массива в новосозданный список | 28  42  58 57 53 51  63  72  80 88  91 | 28  42  58 57 53 51  63  72  80 88  91 |

Код программы

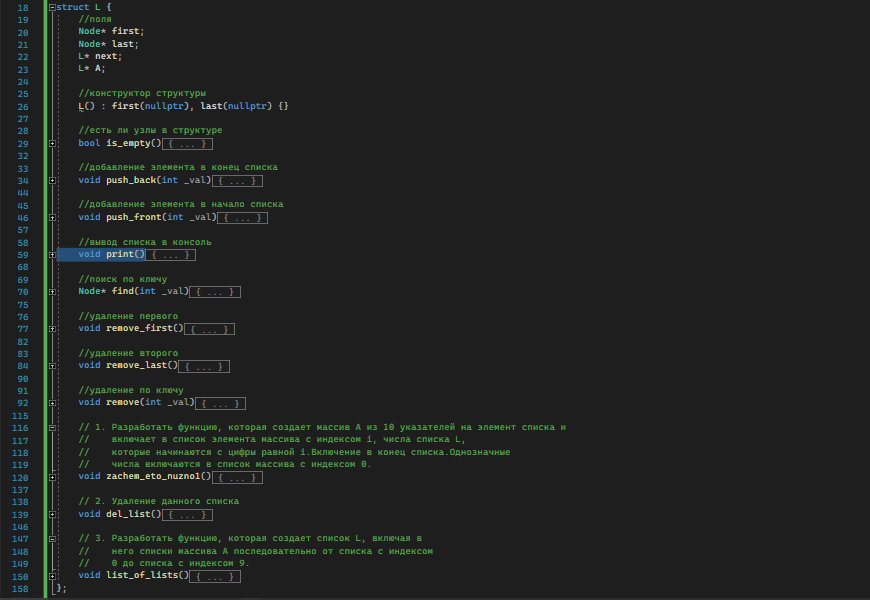


Рисунок 11 – Структуры List и Node



Рисунок 12 – Методы и функции структуры

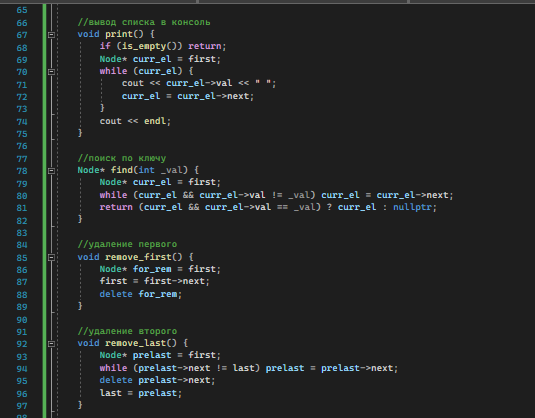


Рисунок 13 – Методы и функции структуры Продолжение

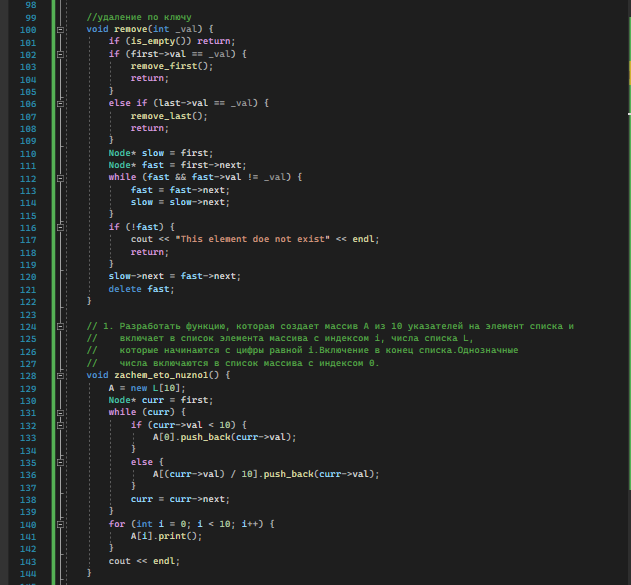


Рисунок 14 – Методы и функции структуры Продолжение

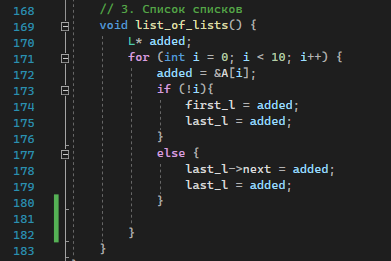
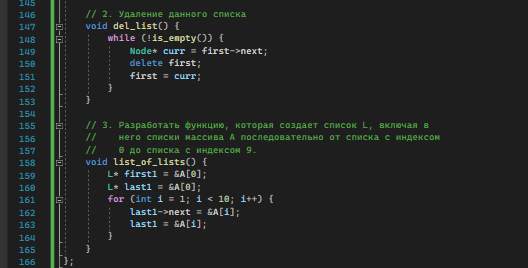


Рисунок 15 – Методы и функции структуры Продолжение

Рисунок 15 – Методы и функции структуры Продолжение

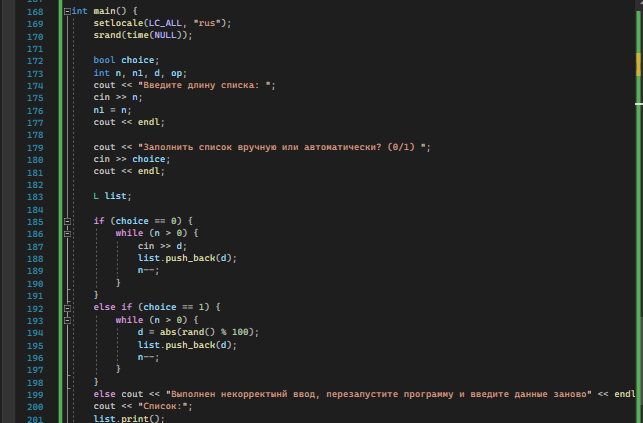


Рисунок 16 – Функция main. Часть 1

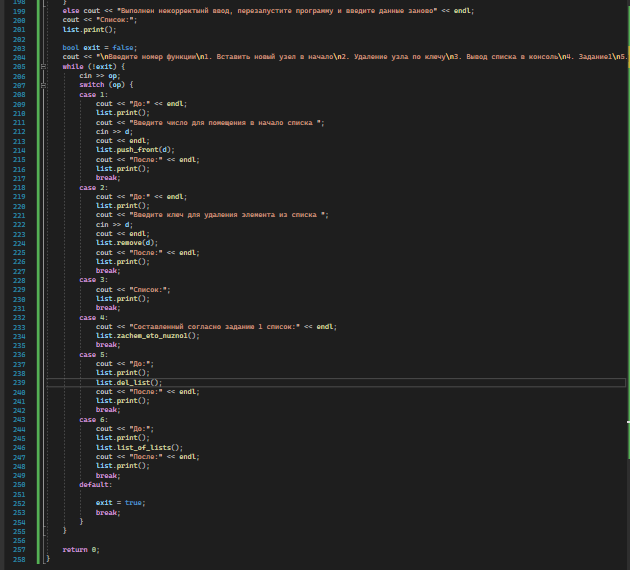
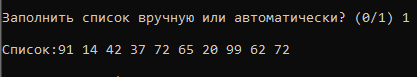


Рисунок 17 – Функция main. Часть 2

Тестирование



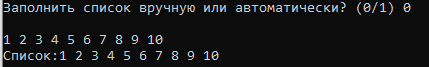
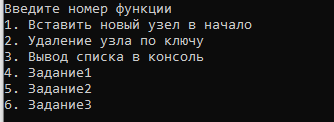
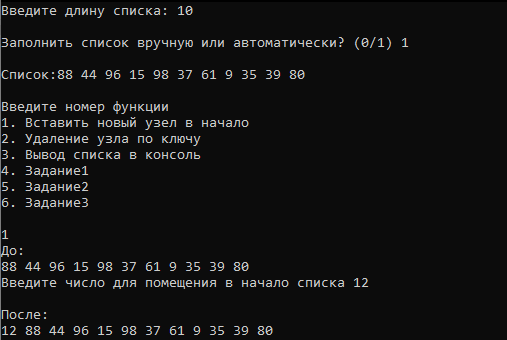
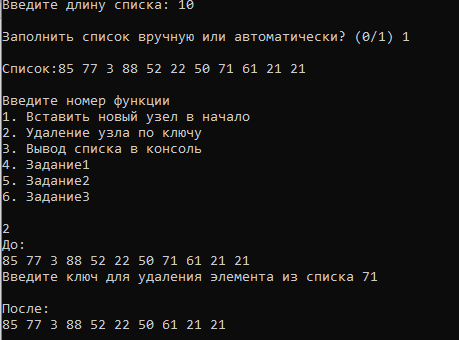


Рисунок 20 – Пользовательское меню

Рисунок 19 – Ручной ввод списка

Рисунок 18 – Выбор автоматической генерации списков

Рисунок 21 – Вставка узла перед первым



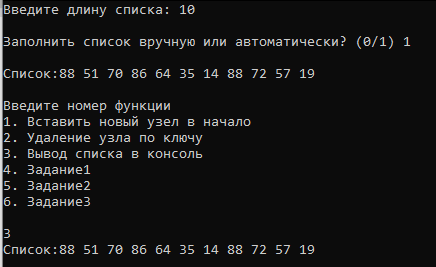


Рисунок 22 – Удаление узла по ключу

Рисунок 23 – Вывод списка в консоль

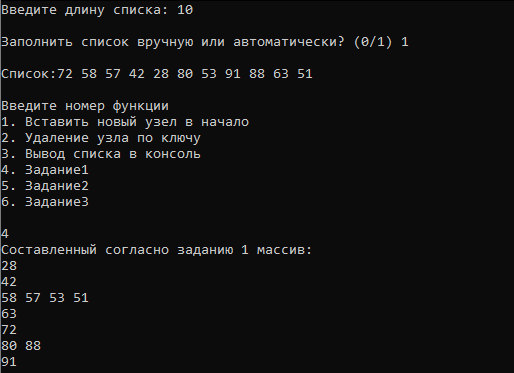




Рисунок 25 – Удаление изначального списка

Рисунок 24 – Создание массива по заданию 1 персонального варианта

Рисунок 25 – Создание списка из списков массива созданного по заданию 1

Вывод

В ходе данной работы были изучены принципы построения и работы линейной динамической структурой данных – односвязный список. Был реализован функционал вставки, удаления, поиска элементов, сортировки списка, формирования нового списка по заданному признаку, удаления подсписка, заданного диапазоном позиций. Результаты тестирования подтвердили корректность выполнения всех операций.