

**Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный
университет»**

Кафедра технологий
программирования

Алгоритмы и структуры данных
Отчет по лабораторной работе №1
Вариант 11

Выполнил

Ланцев Евгений Николаевич.
21-ИТ-1, ФИТ

Проверил

преподаватель
Виноградова А.Д.

Полоцк
2022 г.

Лабораторная работа № 1

“Реализация линейной структуры данных «Список» и основные алгоритмы обработки.”

Цель работы: ознакомиться с основами линейной структуры данных «Список», изучить основные алгоритмы обработки ЛСД «Список», научиться применять полученные знания на практике.

Вариант 11

```
✓ class Node {  
✓     constructor(data, previous, next) {  
        this.data = data;  
        this.previous = previous;  
        this.next = next;  
    }  
}
```

Рисунок 1 - Класс узла

```
class LinkedList {  
    constructor(data) {  
        this.header = new Node(data, null, null);  
    }  
}
```

Рисунок 2 - Класс двусвязного списка

```

find(data) {
    let currentNode = this.header; // корень
    while (currentNode.data !== data) {
        currentNode = currentNode.next; // пока не совпадёт с data
    }
    return currentNode;
}

getLast() {
    let currentNode = this.header;
    while (currentNode.next !== null) {
        // пока не будет указывать на пустой
        currentNode = currentNode.next;
    }
    return currentNode;
}

add(data) {
    let lastNode = this.getLast();
    let newNode = new Node(data, lastNode, null);
    lastNode.next = newNode;
}

remove(data) {
    let currentNode = this.find(data);
    if (currentNode.next !== null && currentNode.previous !== null) {
        currentNode.previous.next = currentNode.next;
        currentNode.next.previous = currentNode.previous;
        currentNode = null;
    } else if (currentNode.next === null) {
        currentNode.previous.next = null;
        currentNode = null;
    } else if (currentNode.prev === null) {
        // удалить корневой
        currentNode.next.previous = null;
        this.header = currentNode.next;
    }
}

```

```

removeSame() {
    // проверяем каждый элемент со всеми для поиска одинаковых элементов
    let currentNode = this.header;
    let nextNode = this.header;
    let x = 0;
    while (currentNode !== null) {
        nextNode = this.header;
        while (nextNode !== null) {
            if (nextNode !== currentNode && nextNode.data === currentNode.data) {
                this.remove(nextNode.data);
            }
            nextNode = nextNode.next;
        }
        currentNode = currentNode.next;
    }
}

display() {
    let currentNode = this.header;
    while (currentNode !== null) {
        console.log(currentNode.data);
        currentNode = currentNode.next;
    }
}

```

Рисунок 3 - Базовые методы

ADD ELEMENT : 1

1-ADD NODE
 2-DISPLAY ALL NODES
 3-DELETE BY NAME
 4-DELETE SAME NODES
 4
 3

1-ADD NODE
 2-DISPLAY ALL NODES
 3-DELETE BY NAME
 4-DELETE SAME NODES
 0-EXIT
 CHOOSE TASK : █

Рисунок 4 - Результат работы программы