

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего
образования



НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

Списки. Стеки. Очереди
Отчет по лабораторной работе
Вариант № 17

по дисциплине

Алгоритмы и структуры данных

РУКОВОДИТЕЛЬ:

_____ Капранов С.Н.

СТУДЕНТ:

_____ Сапожников В.О.

19-ИВТ-3

Работа защищена «___» _____

С оценкой _____

Текст задания

Сделать определённый элемент в списке первым.

Используемый язык программирования: Java

Текст программы

SinglyLinkedList.java

```
/**
 * Реализация односвязного списка.
 *
 * Данная реализация имеет конструктор по умолчанию, создающий
 * пустой список.
 *
 * Классы в java не требуют реализации деструктора, т.к.
 * динамическая память освобождается
 * при помощи "сборщика мусора".
 *
 * Может содержать все элементы любого типа (включая {@code
 * null})
 *
 * @author Vladislav Sapozhnikov
 * @param <E> тип, содержащийся в данной коллекции
 */
public class SinglyLinkedList<E>
{
    /**
     * Класс - узел/ячейка списка.
     * Стандартный класс для большинства динамических структур.
     *
     * @param <E> тип, содержащийся в данном узле
     */
    private static class Node<E>
    {
        E value;                //поле, хранящее данные
        Node<E> next;           //указатель на след. узел

        Node(Node<E> next, E value)
        {
            this.value = value;
            this.next = next;
        }
    }

    private Node<E> head;       //указатель на начало списка
```

```

/**
 * private метод для получения узла по заданному списку.
 *
 * @param index          - индекс, по которому
 *                          необходимо получить
 *                          узел.
 *
 * @throws IndexOutOfBoundsException - исключение,
 *                                    сообщающее о выход за
 *                                    пределы списка.
 *                                    Данное исключение не
 *                                    обрабатывается внутри
 *                                    функции, а
 *                                    делегируется
 *                                    методу, из которого
 *                                    вызвана данная
 *                                    функция.
 *
 * @return Node<E>       - возвращает узел
 *                          списка.
 * */
private Node<E> getByIndex(int index)
{
    if (index < 0)
    {
        throw new IndexOutOfBoundsException("Index must be a
                                             + positive number.");
    }

    Node<E> buffNode = head;
    for(int i = 0; i < index; i++)
    {
        if (buffNode.next == null)
        {
            throw new IndexOutOfBoundsException("Index [" +
                                                  + index + "] +
                                                  + out of +
                                                  + list.");
        }
        buffNode = buffNode.next;
    }
    return buffNode;
}

```

```

public SinglyLinkedList()
{
    head = null;
}

/**
 * Метод, добавляющий новый узел в конец списка.
 *
 * @param value - данные, которые следует добавить в новый
 *              узел списка.
 * */
public void push_back(E value)
{
    if (head == null)
    {
        head = new Node<>(null, value);
    }
    else
    {
        Node<E> buffNode = head;
        while(buffNode.next != null)
        {
            buffNode = buffNode.next;
        }
        buffNode.next = new Node<>(null, value);
    }
}

/**
 * Метод, добавляющий новый узел в начало списка.
 *
 * @param value - данные, которые следует добавить в новый
 *              узел списка.
 * */
public void push_front(E value)
{
    if (head == null)
    {
        head = new Node<>(null, value);
    }
    else
    {
        head = new Node<>(head, value);
    }
}

```

```

/**
 * Перегруженный метод от класса Object.
 * Отвечает за вывод объекта в консоль.
 *
 * @return String - строка, содержащая строковое
 *                  представление объекта.
 * */
@Override
public String toString()
{
    StringBuilder result = new StringBuilder("[");

    Node<E> buffNode = head;

    while(buffNode.next != null)
    {
        result.append(buffNode.value).append(", ");
        buffNode = buffNode.next;
    }
    result.append(buffNode.value).append("]");

    return new String(result);
}

/**
 * Метод, удаляющий элемент после переделанного индекса.
 *
 * @param index - индекс эл-та после которого стоит
 *               произвести удаление.
 * */
public void removeAfter(int index)
{
    Node<E> buffNode = getByIndex(index);
    Node<E> toDelete = buffNode.next;
    buffNode.next = toDelete.next;
}

/**
 * Метод, возвращающий значение из узла по переданному
 * индексу.
 *
 * @param index - индекс узла, содержимое которого
 *               необходимо получить.
 * */

```

```
public E get(int index)
{
    Node <E> buffNode = getByIndex(index);
    return buffNode.value;
}}
```



```

        System.out.println("\nИсходный список: ");
        System.out.println(list);

        System.out.print("\nУкажите индекс элемента для +
                           + перестановки: ");
        String str = inputWithCheck(reader);
        System.out.println("\nИзмененный список: ");
        makeListElFirst(list, str.equals("q") ? 0 :
Integer.parseInt(str));          //ввод пользователем
                                   //индекса для замены
        System.out.println(list);
    }
    catch (IOException e)
    {
        System.err.println("Произошла ошибка ввода. +
                           + Попробуйте еще раз.");
    }
    finally {
        System.out.println("\n\u001B[31m\t\t\t\t\t +
                           + Завершение работы..." +
                           + "\u001B[0m");
    }
}

```

```

/**
 * Функция, проверяющая вводимую строку на содержание
 * только цифр.
 *
 * @param reader      - буферизированный поток ввода.
 *
 * @throws IOException - при неудачной попытке ввода
 *                       бросается исключение потока
 *                       ввод/вывода./
 *
 *                       Данное исключение не обрабатывается
 *                       внутри функции, а делегируется
 *                       методу, из которого вызвана данная
 *                       функция.
 *
 * @return str        - строка с введенными значениями.
 * */

```



```

    public static String inputWithCheck(BufferedReader reader)
    throws IOException
    {
        String str;

        while (true)
        {
            str = reader.readLine();

            if (str.equals("q"))          //выход при введении 'q'
            {
                break;
            }
            else if (!str.matches("-?\\d+(\\.\\d+)?"))
//проверка на содержание только чисел
//при помощи регулярного выражения
            {
                System.err.println("Некорректный ответ. +
                                   + Повторите ввод.");
                continue;
            }
            break;
        }
        return str;
    }

/**
 * Функция, заполняющая переданный лист значениями типа
 * Integer.
 *
 * @param list<Integer> - лист типа Integer для заполнения.
 *
 * @throws IOException - при неудачной попытке ввода
 *                       бросается исключение потока
 *                       ввод/вывода.
 *
 *                       Данное исключение не
 *                       обрабатывается внутри функции, а
 *                       делегируется
 *                       методу, из которого вызвана данная
 *                       функция.
 * */

```

```

        public static void
        fillIngerListManually(SinglyLinkedList<Integer> list,
        BufferedReader reader) throws IOException
        {
            String str;

            while (true)
            {
                System.out.print("Введите след. элемент: ");
                str = inputWithCheck(reader);

                if (str.equals("q")) break;

                list.push_back(Integer.parseInt(str));
            }
        }

        /**
         * Функция, делающая элемент списка по переданному индексу
         * первым в списке.
         * Данная функция реализована только при помощи методов
         * класса SinglyLinkedList
         *
         * @param index    - индекс, по которому будет происходить
         *                  замена.
         * @param list<E> - лист обобщенного типа, в котором будут
         *                  происходить изменения.
         * */
        public static <E> void makeListElFirst(SinglyLinkedList<E>
        list, int index)
        {
            if (index == 0)
            {
                return;
            }
            E value = list.get(index);
            list.removeAfter(index-1);
            list.push_front(value);
        }
    }

```

Скриншоты

Обработка неверных введенных данных.

```
vladislav@progerSapog:~/Документы/Java/Draft for lab1/out/production/Draft for lab1$ java Main

Начало работы...

Для прерывания ввода введите - 'q'
Заполнение списка:
Введите след. элемент: ty
Некорректный ответ. Повторите ввод.
gf
Некорректный ответ. Повторите ввод.
-
Некорректный ответ. Повторите ввод.
12
Введите след. элемент: 4
Введите след. элемент: 346
Введите след. элемент: 45
Введите след. элемент: 4
Введите след. элемент: -
Некорректный ответ. Повторите ввод.
-46
Введите след. элемент: -
Некорректный ответ. Повторите ввод.
-78
Введите след. элемент: q

Исходный список:
[12, 4, 346, 45, 4, -46, -78]

Укажите индекс элемента для перестановки: -1

Измененный список:

Завершение работы...
Exception in thread "main" java.lang.IndexOutOfBoundsException: Index must be a positive number.
```

Работа программы в штатном режиме.

```
vladislav@progerSapog:~/Документы/Java/Draft for lab1/out/production/Draft for lab1$ java Main

Начало работы...

Для прерывания ввода введите - 'q'
Заполнение списка:
Введите след. элемент: 4
Введите след. элемент: 56
Введите след. элемент: 56
Введите след. элемент: -457
Введите след. элемент: -45
Введите след. элемент: -234
Введите след. элемент: 324
Введите след. элемент: -34
Введите след. элемент: q

Исходный список:
[4, 56, 56, -457, -45, -234, 324, -34]

Укажите индекс элемента для перестановки: t
Некорректный ответ. Повторите ввод.
y
Некорректный ответ. Повторите ввод.
h
Некорректный ответ. Повторите ввод.
6

Измененный список:
[324, 4, 56, 56, -457, -45, -234, -34]

Завершение работы...
```