

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

Кафедра технологий  
программирования

Алгоритмы и структуры данных  
Отчет по лабораторной работе №3  
Вариант 11

Выполнил

Ланцев Евгений Николаевич.  
21-ИТ-1, ФИТ

Проверил

преподаватель  
Виноградова А.Д.

Полоцк  
2022 г.

## Лабораторная работа № 3

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

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

**ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ** (ответы на контрольные вопросы):

1. Определение понятия очередь.

Очередью называется упорядоченный набор элементов, которые могут удаляться с её начала и помещаться в её конец.

2. Опишите принцип работы очереди.

Очередь в программировании используется, как и в реальной жизни, когда нужно совершить какие-то действия в порядке их поступления, выполнив их последовательно.

3. Операции для работы с очередью.

- `init()` инициализация очереди.
- `insert (q, x)` — помещение элемента `x` в конец очереди `q` (`q` — указатель на очередь);
- `x=remove (q)` — удаление элемента `x` из очереди `q`;
- `isempty(q)` — возвращает 1, если очередь пуста и 0 в противном случае;
- `print(q)` – вывод элементов очереди `q`.

4. Перечислите способы реализации очереди.

- с помощью одномерного массива;
- с помощью связанного списка;
- с помощью класса объектно-ориентированного программирования.

## Вариант 11

```
const stack = [];
```

Рисунок 1 - очередь

```

"\n1-ADD NODE\n2-DISPLAY ALL NODES\n3-REMOVE FIRST AND LAST\n4-REM
(task) => {
  if (task === 1) {
    // занесение в очередь с учётом приоритета
    rl.question("\nADD ELEMENT : ", (dataElement) => {
      rl.question("\nELEMENT PRIORITET: ", (index) => {
        stack.splice(index, 0, dataElement);
        stack = stack.filter((el) => {
          return el !== null && el !== "";
        });
        tasks();
      });
    });
  }
  else if (task === 2) {
    console.log("\nDISPLAY : \n");
    console.log(stack);
    tasks();
  }
  else if (task === 3) {
    stack.shift();
    stack.pop();
    tasks();
  }
  else if (task === 4) {
    rl.question("\n REMOVE ELEMENT : ", (dataElement) => {
      for (let i = 0; i < stack.length; i++) {
        console.log(stack.length);
        if (stack[i] === dataElement) {
          stack.splice(i, 1);
        }
      }
    });
  }
}

```

Рисунок 2 - Базовые методы для работы с очередью

1-ADD NODE  
2-DISPLAY ALL NODES  
3-REMOVE FIRST AND LAST  
4-REMOVE SPECIFIC ELEMENT  
CHOOSE TASK : 1

ADD ELEMENT : 2

ELEMENT PRIORITET: 4

[]

1-ADD NODE  
2-DISPLAY ALL NODES  
3-REMOVE FIRST AND LAST  
4-REMOVE SPECIFIC ELEMENT  
CHOOSE TASK :

*Рисунок 3 - Результат работы программы*