

Лабораторная работа № 9

Абстрактный тип данных «Очередь»

Очередь (queue) – это другой вид линейной структуры данных. В очереди элементы данных упорядочены от добавленных первыми (начало очереди) к добавленным последними (конец очереди). Элементы данных вставляются в конец очереди и удаляются из начала очереди. Для добавления элементов используется метод enqueue, а для удаления – метод dequeue. Пример применения методов enqueue и dequeue приводится ниже:

enqueue <i>a</i>	enqueue <i>b</i>	enqueue <i>c</i>	dequeue	dequeue
<i>a</i> ← начало	<i>a b</i> ← начало	<i>a b c</i> ← начало	<i>b c</i> ← начало	<i>c</i> ← начало

Движение элементов данных через очередь отражает принцип «первым пришел-первым ушел» (“first in, first out” или FIFO), который характерен при движении покупателей в очереди или перемещении информации через каналы данных.

Класс Queue включает в себя следующие свойства и методы:

```
const int defMaxQueueSize = 10; // максимальный размер очереди по умолчанию
template < class DT >
class Queue
{
public:
    Queue ( int maxNumber = defMaxQueueSize ); // конструктор
    ~Queue (); // деструктор
    // операции с данными в очереди
    void enqueue ( const DT &newData ); // поместить в очередь элемент
    DT dequeue (); // извлечь элемент из очереди
    void clear (); // очистить очередь
    // статус очереди
    bool isEmpty () const; // очередь пуста?
    bool isFull () const; // очередь заполнена?
    int getLength () const; // длина очереди
    void show () const; // вывести данные в очереди на консоль
private:
    // элементы данных
    int maxSize, // максимальное число элементов данных в очереди
        front, // индекс начального элемента данных
        rear; // индекс конечного элемента данных
    DT *dataItems; // массив, содержащий элементы данных
};
```

Особенности реализации класса Queue:

- Конструктор создает пустую очередь и динамически выделяет массив для хранения maxNumber элементов данных.
- Деструктор освобождает память, выделенную в конструкторе.

- Метод `enqueue` добавляет элемент данных в конец очереди, если очередь не заполнена полностью. Если в момент добавления нового элемента данных индекс конца очереди `rear` указывает на последний элемент выделенного массива (т.е. равен `maxSize-1`), то новый элемент данных записывается в начало массива, а индекс конца очереди `rear` устанавливается в значение ноль.
- Метод `dequeue` удаляет элемент данных, добавленный в очередь первым (элемент в начале очереди), и возвращает его, если очередь не пуста.
- Метод `getLength` возвращает количество элементов данных в очереди в текущий момент или ноль, если очередь пуста.

Задание 9.1. (5 баллов) Разработайте конструктор, деструктор, методы `enqueue`, `dequeue`, `clear`, `isEmpty`, `isFull`, `getLength` класса `Queue` с использованием шаблонов классов и с учетом указанных выше особенностей. Разработайте на базе присланного шаблона интерактивную программу для работы с очередью из символов (тип `char`), поддерживающую следующие команды с консоли:

Команда	Действия
<code>+ x</code>	Поместить элемент x в очередь (в конец)
<code>-</code>	Извлечь элемент из очереди и вывести его на консоль
<code>E</code>	Вывести, является ли очередь пустой
<code>F</code>	Вывести, является ли очередь заполненной
<code>#</code>	Вывести на консоль длину очереди
<code>C</code>	Очистить очередь
<code>Q</code>	Выйти из программы

После выполнения каждой команды выводите на консоль состояние очереди при помощи метода `show`.

При тестировании разработанного кода исполните следующие команды:

- `+ a + b + c + d` (добавление элементов в очередь)
- `#` (вывод длины очереди)
- `- - -` (извлечение элементов из очереди)
- `#` (вывод длины очереди)
- `+ e + f + g` (добавление элементов в очередь)
- `E F` (очередь пуста или заполнена?)
- `C` (очистка очереди)
- `E` (очередь пуста?)
- `Q` (выход)

Напоминание

Выполненные задания отправляются со своей электронной почты на почту `labs_it@mail.ru`. Тема письма содержит номер группы, фамилию студента и номер задания, например:

НПИ-01-18 Иванов 4.1

В письмо включается текст программы, скриншот всего экрана с результатом работы программы (с системной датой и временем), пояснения (при необходимости).

Если Вы не успели выполнить задание полностью во время лабораторной работы, то в конце занятия Вы должны отправить на почту файл с текущим состоянием проекта (программы) и указанием выполненных и невыполненных работ. В этом случае сохраняется прежний порядок оценки работы (задание оценивается с максимальной оценкой, если программа корректно решает поставленную задачу и задание отправлено в ходе занятия или в день проведения занятия).

Если в конце занятия файл с программой на почту не поступил или за время лабораторной работы никакие работы не выполнены, то задание оценивается исходя из 75% своей максимальной оценки.