ФИО: Крыжановский Никита Эдуардович

Вариант 4

1. Общая постановка задачи

Разработать программу, реализующую систему работы с очередью, реализованной через массив.

Должна иметь функции

- Вставки элемента  
- Удаления элемента  
- Проверки пустая/полная ли очередь

2. Описание разработанных классов

Класс QueueVector является унаследованным от базового класса Queue и реализует работу очереди.

Методы класса:

- void enQueue(const T& element) – метод для добавления элемента в очередь.

- T deQueue() – метод для удаления элемента из очереди. Должен возвращать удаленный элемент.

- bool isEmpty() – проверяет заполнена ли очередь.

- bool isFull() – проверяет пуста ли очередь.

Должна быть реализована функция для анализа правильности расстановки скобок. Функция должна возвращать True, если количество открывающих и закрывающих скобок одного типа совпадает, и они имеют правильную вложенность. Допускаются три вида скобок: круглые, квадратные и фигурные.

3. Должна присутствовать обработка ошибок

3.1 При попытке удалить элемент из пустой очереди, вывод: “QueueUnderflow”

3.2 При попытке создать очередь размером 0 или меньше вывод: “WrongQueueSize”

**Таблица с детальными требованиями и тест-планом**

|  |  |  |  |
| --- | --- | --- | --- |
| Требование | Детальные требования | Данные | Ожидаемый результат |
| 1.Должна присутствовать | | | |
| 1.1 Обработка ошибки при удалении элемента из пустой очереди | При попытке удалить элемент из пустой очереди, вывод: “QueueUnderflow” | queue.deQueue() | QueueUnderflow |
| 1.3 Обработка ошибки при попытке создать стек недопустимого объема | При попытке создать стек размером 0 или меньше вывод: “WrongQueueSize” | QureueVector<…> queue(0); | WrongQueuSsize |

**Пример работы кода:**

