

Домашнее задание

Есть код реализации абстрактного типа данных «Очередь» на языке программирования C#. Необходимо добавить документацию к коду.

Файл кода на C#

```
package abstractDataType;

// класс АД «Очередь»
public class MyQueueWithChanges<T> {
    private int sIzE = 0; // размер
    private int idx_Start = 1; // индекс начала
    private int idx_end = idx_Start; // индекс конца
    private T[] Array; // массив элементов

    // конструктор класса
    @SuppressWarnings("unchecked")
    MyQueueWithChanges() {
        this.Array = (T[]) new Object[2];
    }

    // метод: получить следующий индекс
    // переменная метода: idx – индекс
    // метод возвращает: индекс
    private int getNextIdx(int idx) {
        if (sIzE == 1)
            return idx_end;
        return idx == Array.length - 1 ? 0 : idx + 1;
    }

    // метод: получить индекс для вставки
    // переменная метода: idx – индекс
    // метод возвращает: индекс
    private int getNextIdxForPush(int idx) {
        if (sIzE == 0)
            return idx_end;
        return idx == Array.length ? 0 : idx + 1;
    }

    // метод: изменить размер
    @SuppressWarnings("unchecked")
    private void reSize() {
        int length = Array.length;
        if (sIzE == length - 1) { //с учетом пустого элемента
            T[] newArray = (T[]) new Object[2 * length];
            int i = 0;
            int j = idx_Start;
            do {
                i++;
            }
        }
    }
}
```

```

        newArray[i] = Array[j];
        j = getNextIdx(j);
    } while (i != sIzE);
    idx_Start = 1;
    idx_end = sIzE;
    Array = newArray;
    print("reSize");
}
}

// метод: добавить в конец
// переменная метода: newElement – элемент
void pushToEnd(T newElement) {
    reSize();
    idx_end = getNextIdxForPush(idx_end);
    Array[idx_end] = newElement;
    sIzE++;
    print("pushToEnd");
}

// метод: удалить первый элемент
void popFirst() {
    if (isEmpty()) {
        System.out.println("Очередь пуста!\n");
        return;
    }
    Array[idx_Start] = null;
    sIzE--;
    idx_Start = getNextIdx(idx_Start);
    if (isEmpty()) {
        idx_end = idx_Start;
    }
    print("popFirst");
}

// метод: вернуть первый элемент
// метод возвращает: элемент
T peekFirst() {
    return Array[idx_Start];
}

// метод: размер
// метод возвращает: размер
int Size() {
    return sIzE;
}

// метод: пуста ли очередь
// метод возвращает: true/false
boolean isEmpty() {
    return sIzE == 0;
}

```

```

// метод: есть ли конкретный элемент
// переменная метода: currentElement – элемент
// метод возвращает: true/false
boolean contains(T currentElement) {
    if (sIzE == 0)
        return false;
    if (idx_end >= idx_Start) {
        return isExistInPart(currentElement, idx_Start, sIzE);
    } else {
        return isExistInPart(currentElement, 0, idx_end + 1)
            && isExistInPart(currentElement, idx_Start, Array.length);
    }
}

// метод: есть ли конкретный элемент в части очереди
// переменная метода: currentElement – элемент
// переменная метода: start – индекс
// переменная метода: size – размер
// метод возвращает: true/false
private boolean isExistInPart(T currentElement, int start, int size) {
    for (int i = start; i < size; i++) {
        if (Array[i] == currentElement)
            return true;
    }
    return false;
}

// метод: вывод операции, элементов и размера
private void print(String operation) {
    System.out.println(operation);
    System.out.println("idx_Start = " + idx_Start + ", idx_end = " + idx_end + ", size = "
        + size);
    for (T t : Array) {
        System.out.print(t + " ");
    }
    System.out.println("\n");
}
}

```

Документирование

1. Задокументируйте код по правилам, предложенным генератором документации.
 - a. Все классы, поля и методы класса должны быть задокументированы.
 - b. Все аргументы и возвращаемые значения методов должны быть задокументированы.
 - c. В общем все нужно задокументировать.
2. Сгенерируйте документацию.

Отчетность

1. Создайте репозиторий.
2. Создайте ветку в репозитории.
3. Добавьте изначальный файл в репозиторий.
4. Закомитьте изменения.
5. Смержите ветку в master.
6. Создайте новую ветку в репозитории.
7. Внесите изменения в файл.
8. Закомитьте изменения.
9. Создайте пулл-реквест или мерж-реквест.
10. Назначьте ревьюера пулл-реквеста (raccoonsw) или мерж-реквеста (raccoon_sw).
11. После прохождения ревью вмержите ветку в мастер.

Критерии оценивания

- Тексты описаний
- Названия методов, аргументов, переменных

Полезные ссылки

- [Doxygen](#)
- [Installation](#)
- [Getting started](#)
- [Documenting the code](#)
- [Doxygen usage](#)