Algorithms 8

TEL-RAN
by Starta Institute

Стек и очередь

ПРЕПОДАВАТЕЛЬ

TEL-RAN
by Starta Institute

Фото преподавателя

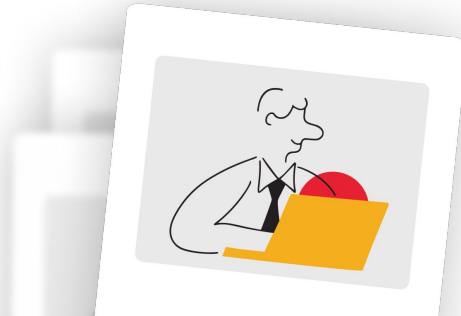
Имя Фамилия

Текущая должность

- Количество лет опыта
- Какой у Вас опыт - ключевые кейсы
- Самые яркие проекты
- Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)



ВАЖНО:

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя.

TEL-RAN
by Starta Institute

ПЛАН ЗАНЯТИЯ

1. Повторение изученного
2. Вопросы по повторению
3. Разбор домашнего задания
4. Основной блок
5. Вопросы по основному блоку
6. Задание для закрепления
7. Практическая работа
8. Оставшиеся вопросы

TEL-RAN
by Starta Institute

TEL-RAN
by Starta Institute

1 ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

TEL-RAN
by Starta Institute

Amortized analysis

- Общие понятия
- Где используется
- Амортизированная стоимость
- Методы AA
- AA операции add в динамический массив

Dynamic Arrays

- Как это работает
- Особенности динамического массива
- Добавить / Удалить
- Изменить размер
- Практика

TEL-RAN
by Starta Institute

2 ВОПРОСЫ ПО ПОВТОРЕНИЮ

TEL-RAN
by Starta Institute

3 РАЗБОР ДОМАШНЕГО ЗАДАНИЯ

Реализация ДЗ

TEL-RAN
by Starta Institute

Реализация на Java

```java
public static void buildDictionary(String text) {
    text = text.toLowerCase();

    int[] result = new int['я' - 'а' + 1];
    for (int i = 0; i < text.length(); i++) {
        char ch = text.charAt(i);
        if (ch >= 'а' && ch <= 'я') {
            result[ch - 'а']++;
        }
    }

    for (int i = 0; i < result.length; i++) {
        System.out.println((char) (i + 'а') + " = " + result[i]);
    }
}
```

Реализация на Java Script

```javascript
function buildDictionary(text) {
    text = text.toLowerCase();

    let result = ['я' - 'а' + 1];
    for (let i = 0; i < text.length; i++) {
        let ch = text.charAt(i);
        if (ch >= 'а' && ch <= 'я') {
            result[ch - 'а']++;
        }
    }

    for (let i = 0; i < result.length; i++) {
        console.log((i + 'а') + " = " + result[i]);
    }
}
```

Введение

TEL-RAN
by Starta Institute

Стек (Stack)

- Стек как структура данных (Stack data structure)
- Методы в Стеке (Methods in Stack)
- Реализация на практике

Очередь (Queue)

- Очередь как структура данных (Queue data structure)
- Методы очереди (Methods in Queue)
- Реализация на практике

TEL-RAN
by Starta Institute

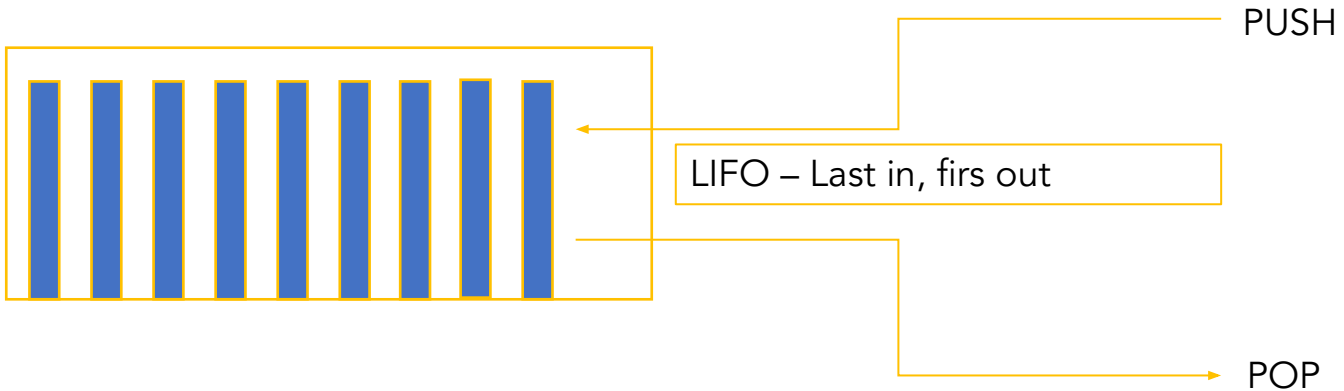
4 ОСНОВНОЙ БЛОК

Структура данных Стек

TEL-RAN
by Starta Institute

Стек — это линейная структура данных, которая следует определенному порядку выполнения операций.

Порядок LIFO (последним пришел, первым ушел).



Операции в Стеке

TEL-RAN
by Starta Institute

Стек поддерживает следующие операции:

- empty — проверка стека на наличие в нем элементов,
- push — операция вставки нового элемента,
- pop — операция удаления нового элемента.

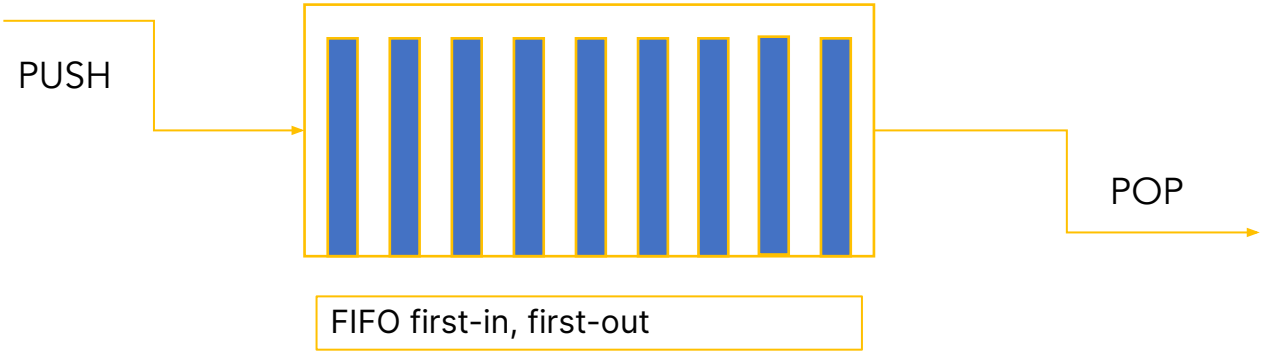
Структура данных Очередь

TEL-RAN
by Starta Institute

Очередь — это линейная структура данных, добавление и удаление элементов в которой происходит путём операций push и pop

Первым из очереди удаляется элемент, который был помещен туда первым, то есть в очереди реализуется принцип «первым пришел — первым ушел» (first-in, first-out — FIFO).

У очереди имеется голова (head) и хвост (tail).



Операции в Очереди

TEL-RAN
by Starta Institute

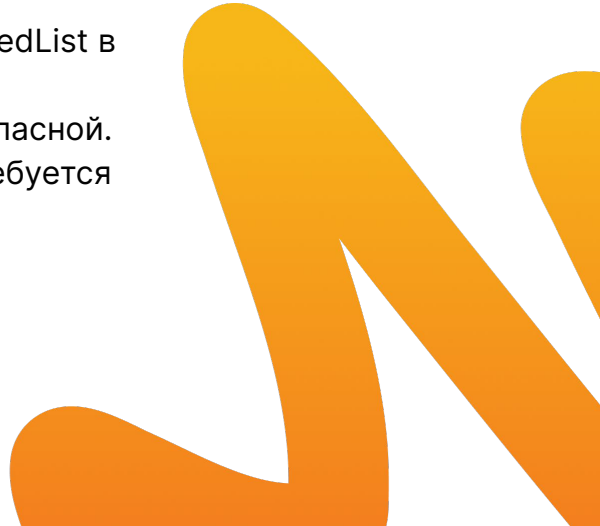
Очередь поддерживает следующие операции:

- empty — проверка очереди на наличие в ней элементов,
- push — операция вставки нового элемента,
- pop — операция удаления нового элемента,
- size — операция получения количества элементов в очереди.

Очередь (Queue)

TEL-RAN
by Starta Institute

- Интерфейс Queue присутствует в пакете java.util и расширяет интерфейс Collection, используемый для хранения элементов, которые должны быть обработаны, в порядке FIFO (первым пришел — первым ушел).
- Это упорядоченный список объектов, использование которого ограничено вставкой элементов в конец списка и удалением элементов из начала списка (FIFO(- First In First Out
- Очередь нуждается в конкретном классе для объявления.
- Наиболее распространенными классами являются PriorityQueue и LinkedList в Java.
- Ни одна из этих реализаций не является потокобезопасной. PriorityBlockingQueue — одна из альтернативных реализаций, если требуется поточно-ориентированная реализация.



Создание объектов Queue

TEL-RAN
by Starta Institute

- Поскольку Queue является интерфейсом, объекты типа Queue не могут быть созданы.
- Нам всегда нужен класс, который расширяет этот список, чтобы создать объект.
- После введения Generics в Java 1.5, можно ограничить тип Queue, который может храниться в Queue.
- Queue<Object> queue = new PriorityQueue<Object>();

Operations on Queue Interface - PriorityQueue

TEL-RAN
by Starta Institute

Добавление элементов:

- Чтобы добавить элемент в очередь, мы можем использовать метод add().
- Порядок размещения не сохраняется в PriorityQueue.
- Элементы сохраняются в порядке приоритета, который по умолчанию является возрастающим.

Operations on Queue Interface - PriorityQueue

TEL-RAN
by Starta Institute

Удаление элементов.

- Чтобы удалить элемент из очереди, мы можем использовать метод remove().
- Если таких объектов несколько, первое вхождение объекта удаляется.
- Кроме того, метод poll() также используется для удаления головы и ее возврата.

Operations on Queue Interface - PriorityQueue

TEL-RAN
by Starta Institute

Итерация очереди.

- Существует несколько способов итерации очереди.
- Самый известный способ — преобразование очереди в массив и обход с помощью цикла for.
- Очередь также имеет встроенный итератор, который можно использовать для перебора очереди.

Выводы

TEL-RAN
by Starta Institute

1. Java Queue используется для вставки элементов в конец очереди и удаления из начала очереди. Он следует концепции FIFO.
2. Java Queue поддерживает все методы интерфейса Collection, включая вставку, удаление и т. д.
3. LinkedList , ArrayBlockingQueue и PriorityQueue — наиболее часто используемые реализации.
4. Если над BlockingQueues выполняется какая-либо пустая операция, генерируется исключение NullPointerException.
5. Все очереди, кроме Deques, поддерживают вставку и удаление в конце и в начале очереди соответственно. Deques поддерживают вставку и удаление элементов с обоих концов.

Экспресс-опрос

TEL-RAN
by Starta Institute

- **Вопрос 1.**

  Объясните FIFO и LIFO

- **Вопрос 2.**

  Предположите где, лучше использовать Стек, а где Очередь?

TEL-RAN
by Starta Institute

5 ВОПРОСЫ ПО ОСНОВНОМУ БЛОКУ

TEL-RAN
by Starta Institute

6 ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ

Задание

TEL-RAN
by Starta Institute

Реализовать Стек и все операции:

Классический Steck - реализовать используя массив (Array)

empty — проверка стека на наличие в нем элементов,
push — операция вставки нового элемента,
pop — операция удаления нового элемента.

ЗАДАНИЕ

TEL-RAN
by Starta Institute

Реализовать нижеперечисленные операции самостоятельно для структуры Стек:

- peek() – Возвращает элемент с вершины стека, но не удаляет его.
- search(element) - Определяет, существует ли объект в стеке.
  Если элемент найден, возвращает позицию элемента с вершины стека. В противном случае он возвращает -1.

TEL-RAN
by Starta Institute

7 ПРАКТИЧЕСКАЯ РАБОТА

Практическое задание 1

TEL-RAN
by Starta Institute

Классическая Queue - реализовать используя массив (Array)

empty — проверка очереди на наличие в ней элементов,

push — операция вставки нового элемента,

pop — операция удаления нового элемента,

size — операция получения количества элементов в очереди.

```
arr;        // массив для хранения элементов queue
head;       // head указывает на первый элемент в queue
tail;       // tail часть указывает на последний элемент в queue
capacity;   // максимальная емкость queue
count;      // текущий размер queue
```

Реализация задания 1

TEL-RAN
by Starta Institute

Реализация на Java

```
public ClassicQueue(int size) {
    arr = new int[size];
    capacity = size;
    head = 0;
    tail = -1;
    count = 0;
}
```

Реализация на Java Script

```
class Queue {
    constructor() {
        this.items = [];
    }

    enqueue(element) {
        this.items.push(element);
    }

    ...
```

TEL-RAN
by Starta Institute

8 ОСТАВШИЕСЯ ВОПРОСЫ

Домашнее задание

TEL-RAN
by Starta Institute

Задача getMin().

Реализовать структуру данных SpecialStack, которая поддерживает все операции со стеком, такие как push(), pop(), isEmpty(), … и дополнительную операцию getMin(), которая должна возвращать минимальный элемент из SpecialStack.

Все эти операции SpecialStack должны быть O(1). Пространство O(n)

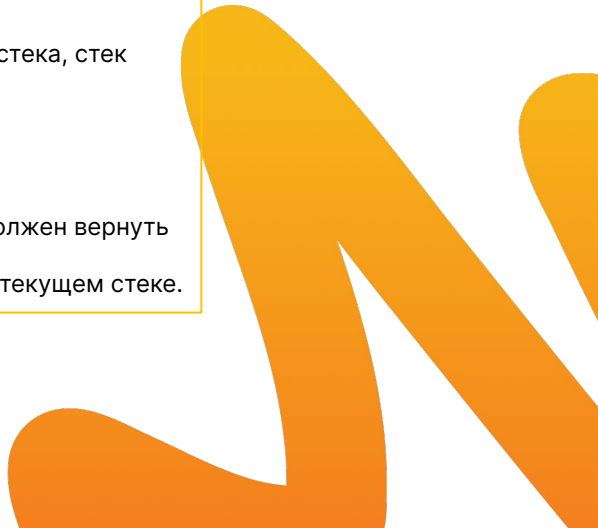
Чтобы реализовать SpecialStack, используйте структуру данных Stack, реализованную ранее в классе.

Рассмотрим следующий SpecialStack
16 → ВЕРХ
15
29
19
18

Когда вызывается getMin(), он должен возвращать 15,
который является минимальным элементом в текущем стеке.

Если мы вытолкнем два раза из стека, стек станет
29 → ВЕРХ
19
18

Когда вызывается getMin(), он должен вернуть 18
который является минимумом в текущем стеке.



Полезные ссылки

TEL-RAN
by Starta Institute

- Stack (abstract data type) - Wikipedia
- Stack (Java SE 10 & JDK 10)
- Queue (abstract data type) - Wikipedia
- Queue (Java Platform SE 8)

ЗАКЛЮЧЕНИЕ

TEL-RAN
by Starta Institute