Java Pro 2.2

TEL-RAN
by Starta Institute

Очередь. Стек. Ассоциативный массив. Класс Collections

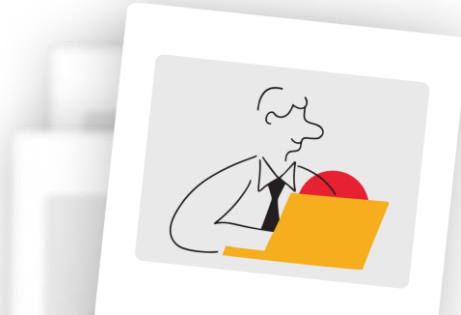
ПРЕПОДАВАТЕЛЬ

TEL-RAN
by Starta Institute

Юрий Костяной

Java/Kotlin backend-разработчик

- 3+ года опыта в коммерческой разработке
- 2+ года опыта в преподавании
- Проекты по интеграции сторонних платформ, CRM
- Проблемно-ориентированный подход в преподавании



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. Основной блок
2. Вопросы по основному блоку
3. Домашняя работа

TEL-RAN
by Starta Institute

TEL-RAN
by Starta Institute

1 ПОВТОРЕНИЕ ИЗУЧЕННОГО

Повторение

TEL-RAN
by Starta Institute

15. Что вы знаете о реализации классов HashSet и TreeSet?
19. Как задается порядок следования объектов в коллекции, как отсортировать коллекцию?
20. Дайте определение понятию “итератор”.
27. В чем разница между Iterator и Enumeration?
31. В чем разница между Iterator и ListIterator?
32. Какие есть способы перебора всех элементов List?
34. Что делать, чтобы не возникло исключение ConcurrentModificationException?

Ответы https://javastudy.ru/interview/collections/

Повторение

Дан класс Car.
1 Какую коллекцию лучше использовать для хранения автомобилей в реестре, если поиск автомобилей в основном происходит по марке и модели автомобиля?
2 Какая коллекция позволит быстро проверять, что такой автомобиль уже есть в коллекции?

```
public class Car {
    private String brand;
    private String model;
    private int engineNumber;
    private BodyType body;

// конструктор, геттеры

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;

        if (engineNumber != car.engineNumber) return false;
        if (!brand.equals(car.brand)) return false;
        if (!model.equals(car.model)) return false;
        return body.equals(car.body);
    }

    @Override
    public int hashCode() {
        int result = brand.hashCode();
        result = 31 * result + model.hashCode();
        result = 31 * result + engineNumber;
        result = 31 * result + body.hashCode();
        return result;
    }
}
```

Повторение

Дан класс Car.
1 Какую коллекцию лучше использовать для хранения автомобилей в реестре, если поиск автомобилей в основном происходит по марке и модели автомобиля?
2 Какая коллекция позволит быстро проверять, что такой автомобиль уже есть в коллекции?

1 TreeSet, т.к. поиск происходит только по двум полям объекта, то применить equals и hashCode для поиска не удастся, поэтому придётся перебирать элементы в коллекции. Поиск в TreeSet будет равен в худшем случае времени поиска в самой длинной ветке.
2 Когда дан объект класса Car, то можно искать по equals и hashCode, т.е. лучше использовать HashSet.

```
public class Car {
    private String brand;
    private String model;
    private int engineNumber;
    private BodyType body;

// конструктор, геттеры

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;

        if (engineNumber != car.engineNumber) return false;
        if (!brand.equals(car.brand)) return false;
        if (!model.equals(car.model)) return false;
        return body.equals(car.body);
    }

    @Override
    public int hashCode() {
        int result = brand.hashCode();
        result = 31 * result + model.hashCode();
        result = 31 * result + engineNumber;
        result = 31 * result + body.hashCode();
        return result;
    }
}
```

Повторение

1 Что нужно, чтобы хранить автомобили в алфавитном порядке марок?
2 Что нужно сделать, чтобы отсортировать список экземпляров по номеру двигателя?

```
public class Car {
    private String brand;
    private String model;
    private int engineNumber;
    private BodyType body;

// конструктор, геттеры

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;

        if (engineNumber != car.engineNumber) return false;
        if (!brand.equals(car.brand)) return false;
        if (!model.equals(car.model)) return false;
        return body.equals(car.body);
    }

    @Override
    public int hashCode() {
        int result = brand.hashCode();
        result = 31 * result + model.hashCode();
        result = 31 * result + engineNumber;
        result = 31 * result + body.hashCode();
        return result;
    }
}
```

Повторение

1 Что нужно, чтобы хранить автомобили в алфавитном порядке марок?
2 Что нужно сделать, чтобы отсортировать список экземпляров по номеру двигателя?

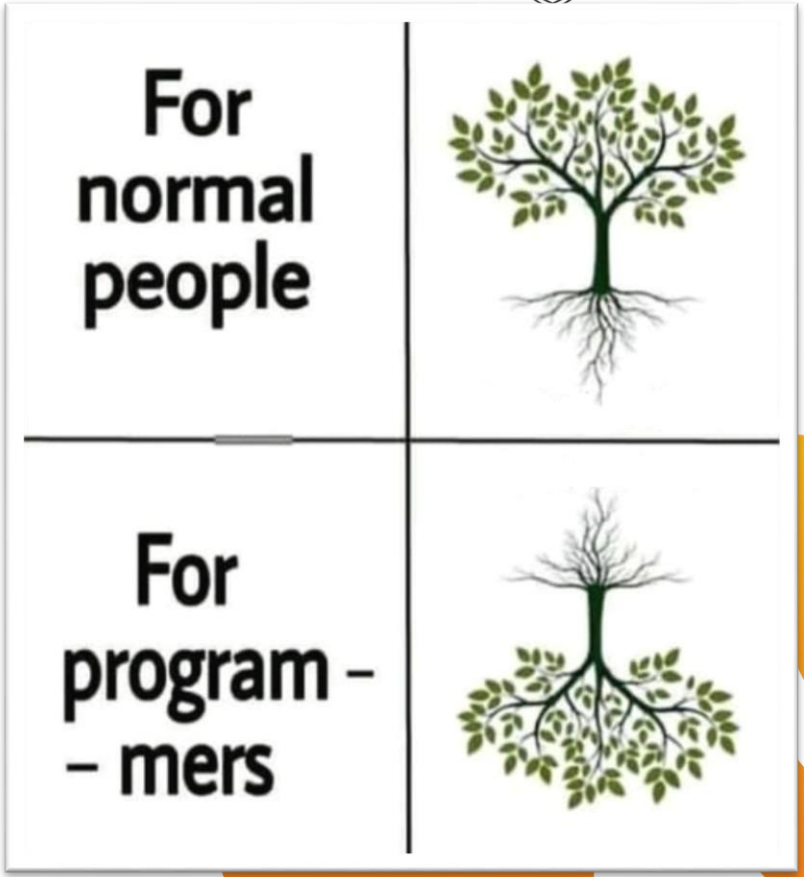
1 Имплементировать интерфейс Comparable.
2 Имплементировать интерфейс Comparator.

```
public class Car {
    private String brand;
    private String model;
    private int engineNumber;
    private BodyType body;

// конструктор, геттеры

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Car car)) return false;

        if (engineNumber != car.engineNumber) return false;
        if (!brand.equals(car.brand)) return false;
        if (!model.equals(car.model)) return false;
        return body.equals(car.body);
    }

    @Override
    public int hashCode() {
        int result = brand.hashCode();
        result = 31 * result + model.hashCode();
        result = 31 * result + engineNumber;
        result = 31 * result + body.hashCode();
        return result;
    }
}
```

Повторение

TEL-RAN
by Starta Institute

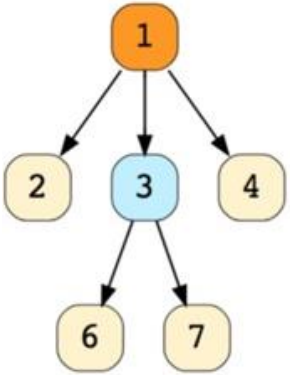
В чём прикол мема?

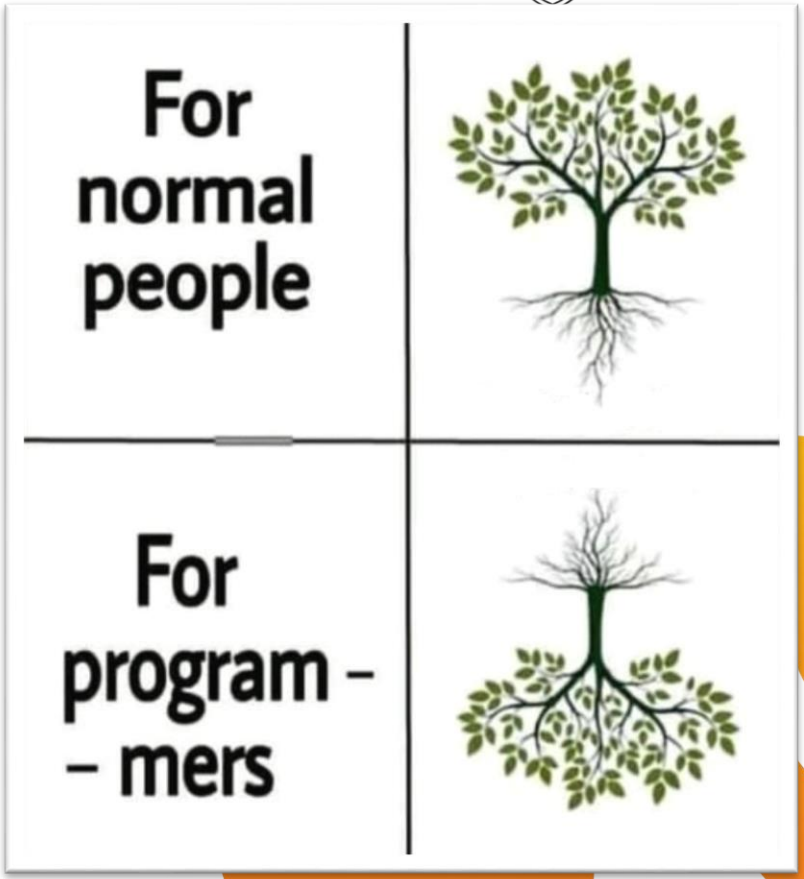


Повторение

TEL-RAN
by Starta Institute

В чём прикол мема?





TEL-RAN
by Starta Institute

2 ОСНОВНОЙ БЛОК

Введение

TEL-RAN
by Starta Institute

- Кто крайний?
- Мне только спросить
- В регистратуру за картой
- Верный помощник коллекций

Проблема

TEL-RAN
by Starta Institute

Ранее мы создали программу для хранения данных о пациентах.

Теперь нужно подумать о модуле программы, который будет отвечать за организацию электронной очереди на приём к доктору.

Кто крайний?

Очередь (queue)

TEL-RAN
by Starta Institute

Queue – список элементов, в котором можно добавлять элементы только в хвост, а удалять – только из начала (*FIFO* - first in, first out) – «первым пришёл, первым ушёл».

Вам обязательно напомнят это правило, если попробуете пролезть без очереди в поликлинике.

Кто крайний?

Методы очереди

TEL-RAN
by Starta Institute

Обладает всеми методами интерфейса *Collection*.

boolean add(E e) или *offer(E e)* – добавление элемента в конец очереди. Если ёмкость очереди превышена, то элемент не будет добавлен и будет выброшено исключение *IllegalStateException*.

boolean addAll(Collection<? extends E> c) – добавляет все элементы коллекции в очередь.

void clear() – удаляет все элементы очереди.

E element() или *peek()* – берёт первый элемент очереди без удаления из очереди.

E remove() или *poll()*– извлекает элемент из очереди.

Кто крайний?

Реализации очереди

TEL-RAN
by Starta Institute

Наиболее популярная реализация очереди – класс *LinkedList*.

The word queue is ironic.

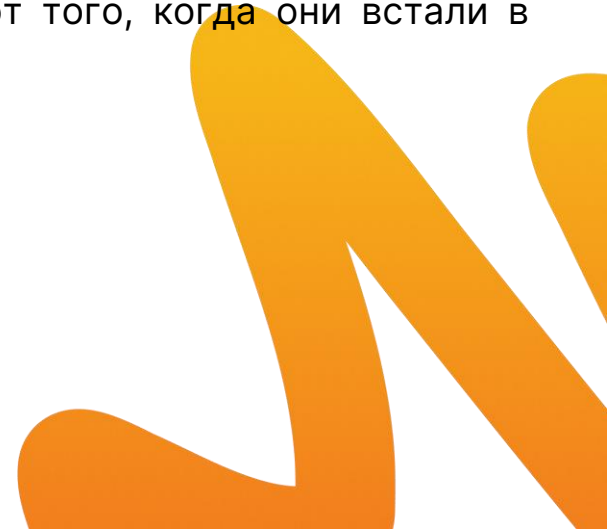
It's just a 'q' with a bunch of silent letters waiting in a line.

Проблема

TEL-RAN
by Starta Institute

В настоящей поликлинике пациенты, забронировавшие время обычно принимаются первыми, а только потом те, кто не имеет бронирования. Кроме того пациенты с острой болью и высокой температурой чащу всего проходят без очереди. Также есть льготные категории граждан, которые имеют право проходить к доктору без очереди (маломобильные люди, пожилые, беременные, дети до года и т.д.). При этом пациенты без записи каждого типа (по живой очереди, с острой болью и льготные) тоже должны иметь порядок в пределах своей группы – в зависимости от того, когда они встали в очередь.

Как можно учесть это огромное количество условий?



Кто крайний?

Реализации очереди

TEL-RAN
by Starta Institute

Класс **PriorityQueue** – упорядоченная очередь. По умолчанию элементы добавляются в естественном порядке: числа по возрастанию, строки по алфавиту и так далее, либо алгоритм сравнения задаёт разработчик.

Этот класс может быть полезен, например, для нахождения n минимальных чисел в большом неупорядоченном списке. Такая реализация выгоднее по скорости и объёму памяти, чем подход с сортировкой первоначального списка.

PriorityQueueExample.zip

Кто крайний?

Задание

TEL-RAN
by Starta Institute

Дополните проект программы для поликлиники классом, который будет формировать очереди пациентов с учётом их приоритетов.

Определите методы добавления пациента в очередь и получения следующего пациента, идущего к врачу.

При необходимости добавьте необходимые поля пациентам.

Проблема

TEL-RAN
by Starta Institute

В реальной жизни многие пациенты обращаются к врачу в том числе за информацией и уточнениями («Мне только спросить»). Поэтому администрация поликлиники решила в помощь каждому врачу назначить медсестру, которая будет принимать таких пациентов.

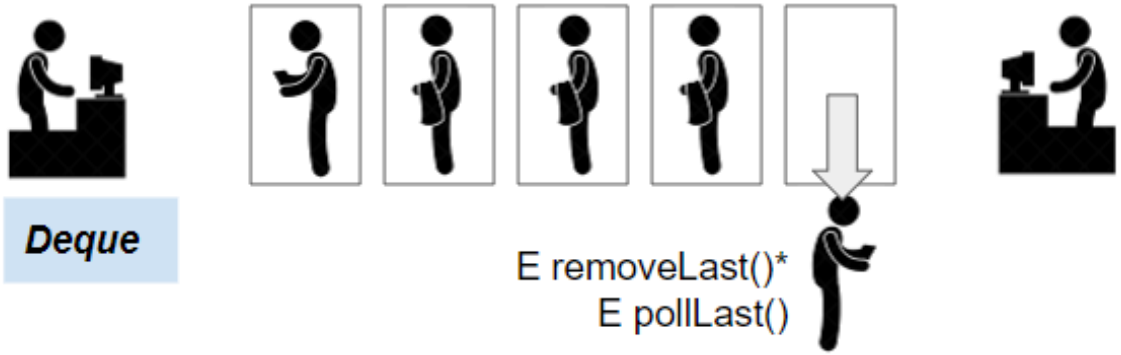
Как организовать очередь с учётом таких пациентов?

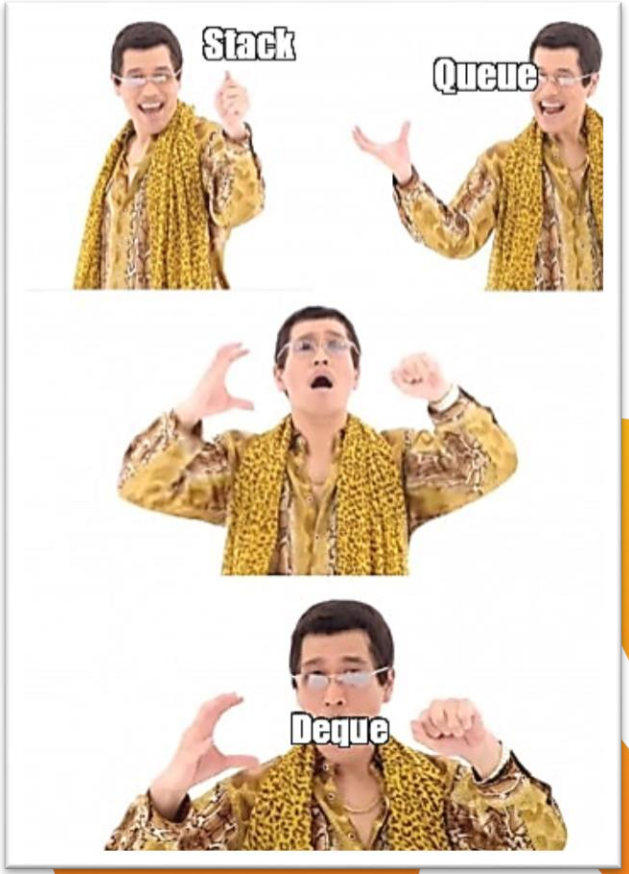
Мне только спросить

Deque (дэк)

TEL-RAN
by Starta Institute

Deque (*double ended queue*, *дэк*, *двостороння очередь*) – это структура данных, которая позволяет добавлять и брать элементы из очереди с двух сторон – начала и конца. Можно сказать, что функционально *дэк* состоит из двух частей – обычной *очереди* и *стека*.



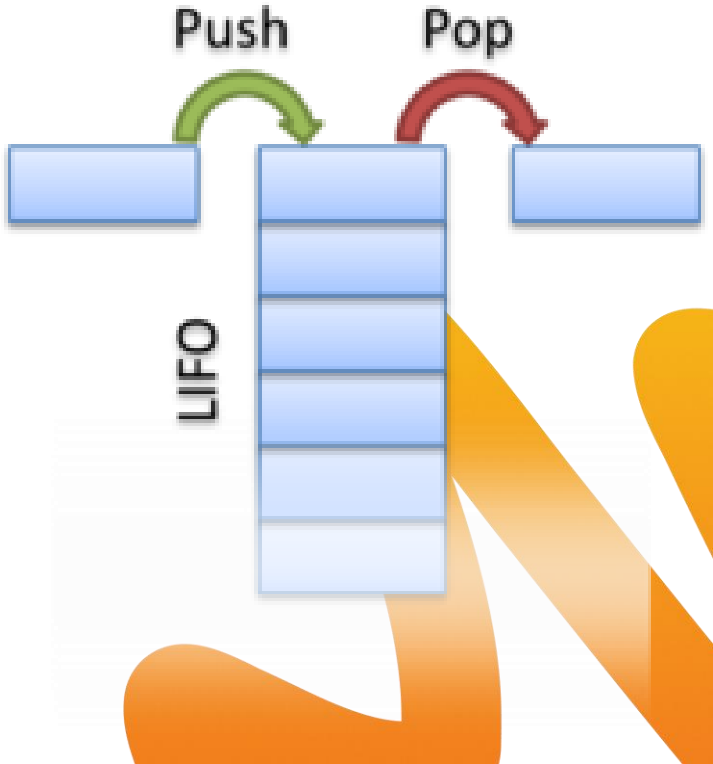


Мне только спросить

Stack (стек)

Стек – это структура данных, организованная по принципу **LIFO** (last in, first out), то есть «последним пришёл — первым ушёл».

Пример – стопка рекламных буклетов на ресепшене отеля: первыми забирают самые верхние (положенные последними).



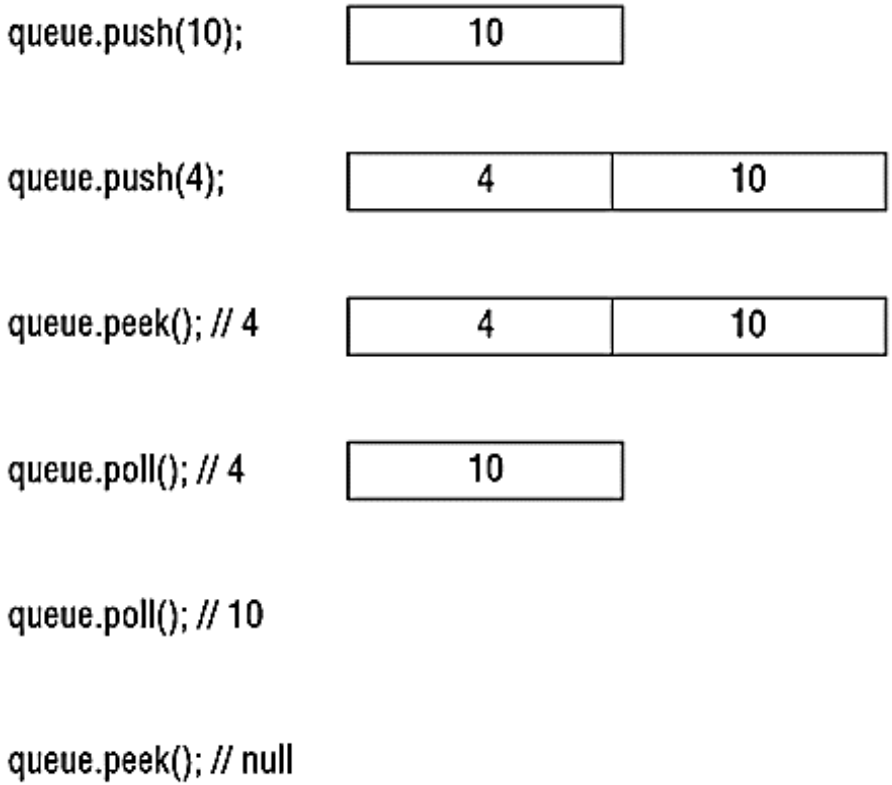
Мне только спросить

Реализация стека

TEL-RAN
by Starta Institute

Когда в приложении нужно создать стек, то обычно используют одну из реализаций *Deque*, игнорируя в ней функционал очереди.

Класс **ArrayDeque** – это реализация двунаправленной очереди в виде массива с переменным числом элементов.



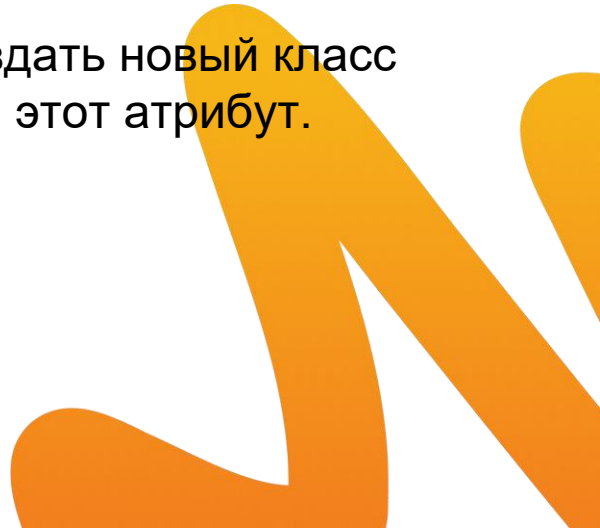
Мне только спросить
Задание

TEL-RAN
by Starta Institute

Дополните класс, формирующий очередь так, чтобы он задавал наименьший приоритет тем, кто хочет спросить.

Добавьте в класс метод, который будет возвращать пациента из конца очереди, если тот хочет «только спросить».

Для хранения поля «мне только спросить» лучше создать новый класс *Место в очереди*, который будет хранить пациента и этот атрибут.



Проблема

TEL-RAN
by Starta Institute

У каждого пациента должна быть медицинская карта с историей болезни.

Конечно, мы можем хранить историю болезни в отдельном поле пациента (композиция). Но тогда изменение одного класса может привести к необходимости менять другой.

Как сделать хранение историй болезни пациентов эффективным?

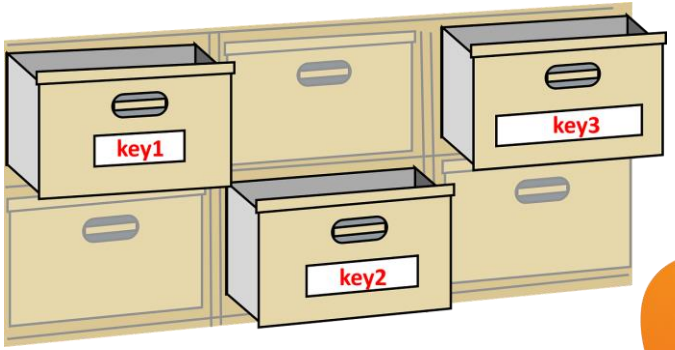
В регистратуру за картой

Map (ассоциативный массив)

TEL-RAN
by Starta Institute

Map (*мапа, карта, словарь, ассоциативный массив*) состоит из пар «ключ-значение». Ключи уникальны (множество), а значения могут повторяться. Порядок элементов не гарантирован. *Map* позволяет искать объекты (значения) по ключу (без перебора всех элементов.

Пример: стопка карточек с иностранными словами и их значениями. Для каждого слова (ключ) на обороте карточки есть вариант перевода (значение), а вытаскивать карточки можно в любом порядке.



В регистратуру за картой

Map (ассоциативный массив)

TEL-RAN
by Starta Institute

Не путайте интерфейс *Collection* и фреймворк *Java Collections Framework (JCF).*

Интерфейс *Map* не наследуется от интерфейса *Collection*, но входит в состав фреймворка *JCF*.

В регистратуру за картой

Методы Map

TEL-RAN
by Starta Institute

Map.of() – метод явного создания неизменяемой мапы.

void clear() – удаление всех объектов мапы.

containsKey(Object key) – проверяет, содержит ли мапа данный ключ.

containsValue(Object value) – проверяет, содержит ли мапа данное значение.

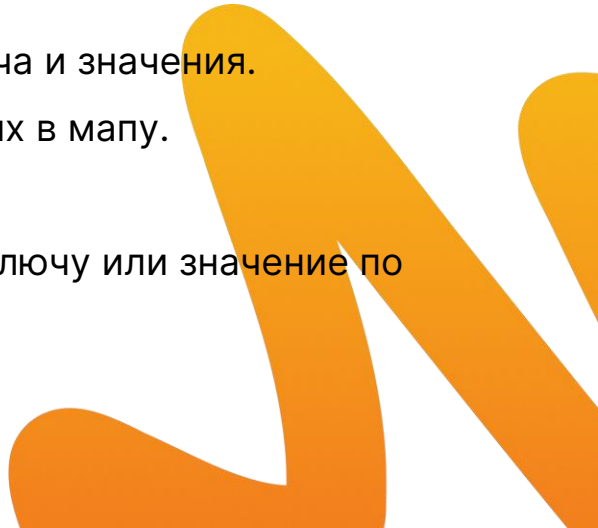
copyOf() – создаёт неизменяемую копию данной мапы.

entry(K k, V v) – возвращает пару ключ-значение из данного ключа и значения.

entrySet() – возвращает множество пар ключ-значение, входящих в мапу.

get(Object key) – получить значение по ключу.

getOrDefault(Object key, V defaultValue) – получить значение по ключу или значение по умолчанию.



В регистратуру за картой

Методы Map

TEL-RAN
by Starta Institute

isEmpty() – проверка, пуста ли мапа.

keySet() – получить множество ключей.

put(K key, V value) – положить данное значение по данному ключу. Если такой ключ есть, то значение по нему заменится новым.

putAll(Map<? extends K,? extends V> m) – моложить все пары ключ-значение из данной мапы в текущую.

putIfAbsent(K key, V value) – положить значение по ключу, если такого ключа ещё нет в мапе.

remove(Object key) – удалить пару ключ-значение по данному ключу

remove(Object key, Object value) – удалить пару ключ-значение, если оба совпали.

В регистратуру за картой

Методы Map

TEL-RAN
by Starta Institute

replace(K key, V value) – заменить значение по ключу

replace(K key, V oldValue, V newValue) – то же, если старое значение есть.

size() – размер мапы

values() – значения мапы в виде коллекции

Полный список методов см. в документации

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Map.html

В регистратуру за картой

Методы Map

TEL-RAN
by Starta Institute

Класс **HashMap** хранит данные в виде хеш-таблицы, как и *HashSet*. Более того, *HashSet* внутри использует *HashMap*. При этом ключом выступает сам элемент.

Класс **TreeMap** строится тоже на базе *красно-чёрного дерева*. Элементы здесь упорядочены (в естественном или заданном при создании порядке) в каждый момент времени. При этом вставка и удаление более затратнее, чем в случае с *HashMap*.

Класс **LinkedHashMap** расширяет возможности *HashMap* тем, что позволяет итерироваться по элементам в порядке их добавления. Как и в LinkedList, здесь каждая пара-значение содержит ссылку на предыдущий и последующий элементы.

Вас много, а я одна!

Задание

TEL-RAN
by Starta Institute

1 Создайте класс История болезни пациента (поля придумайте самостоятельно).

2 Создайте класс для хранения соответствия пациента его истории болезни.

Верный помощник коллекций
Collections

TEL-RAN
by Starta Institute

По аналогии с классом Arrays для массивов существует класс **Collections** для коллекций, который помогает работать с ними. Он предоставляет ряд полезных утилитарных методов для создания пустых коллекций, неизменяемых коллекций, синхронизированных коллекций, методы сортировки коллекций, метод для следования элементов в обратном порядке, поиска элементов в коллекциях и многого другого.

https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Collections.html

TEL-RAN
by Starta Institute

4 Домашнее задание

Домашнее задание

TEL-RAN
by Starta Institute

№1

Создайте программу Вышибала, которая обслуживает посетителей в порядке их прихода в ночной клуб. Если посетителю меньше 21 года, то в клуб его не пустят.

№ 2

Напишите программу Завтрак у бабушки. Бабушка жарит блинчик и кладёт его сверху на стопку. Внук может скушать только верхний блинчик. В цикле смоделируйте, что за одну итерацию бабушка жарит 2 блинчика, а внук съедает только один. Когда внук наелся. Цикл заканчивается. Количество блинов, которые может съесть внук равно его возрасту.

№3

Создайте программу-переводчик с английского языка. Пользователь вводит слово на английском языке, переводчик показывает перевод. Если переводчик не обнаружил в словаре введённое слово, то он просит пользователя ввести перевод, после чего сохраняет слово и его перевод в словарь. Программа продолжается, пока пользователь не введёт stopTranslate.

ЗАКЛЮЧЕНИЕ

TEL-RAN
by Starta Institute