**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра «Математическая Кибернетика и Информационные технологии»

Лабораторная работа №3

Выполнил: Студент группы

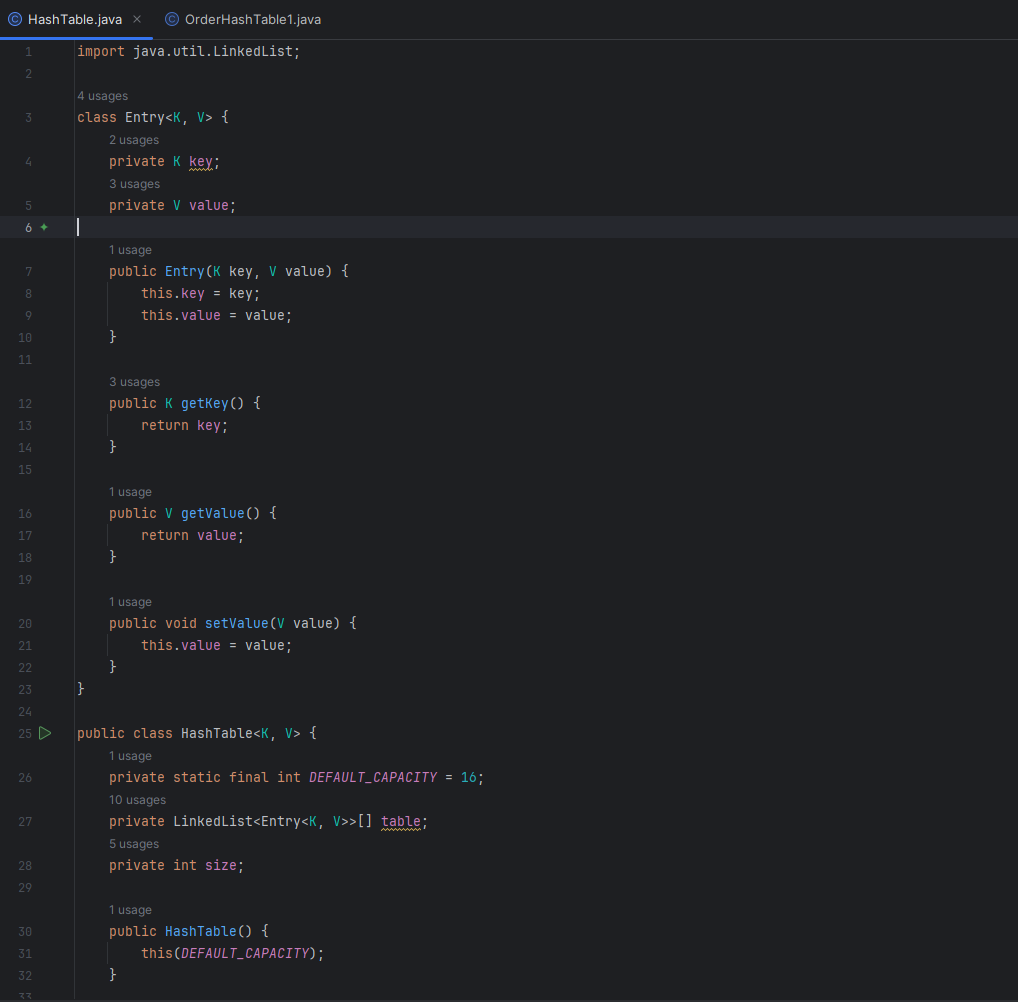
БВТ2201

Бутарасов Дмитрий

Алексеевич

Москва

2023



В данном коде реализуем простую хэш-таблицу с использованием цепочек для решения коллизий.

Entry<K, V> - это класс, который будет храниться в хэш-таблице ключ и значение.

HashTable<K, V> - это основной класс, представляющий саму хэш-таблицу.

private static final int DEFAULT\_CAPACITY = 16; - устанавливает стандартную начальную емкость хэш-таблицы.

private LinkedList<Entry<K, V>>[] table; - это массив списков, представляющих корзины (buckets) хэш-таблицы. Каждая корзина содержит связанный список элементов с одинаковым хэшем.

private int size; - отслеживает текущий размер хэш-таблицы.

public HashTable() - конструктор по умолчанию, который вызывает конструктор с параметром, устанавливая начальную емкость в значение по умолчанию.

public HashTable(int capacity) - конструктор, который создает массив корзин (table) с указанной начальной емкостью.

Класс HashTable содержит методы для основных операций:

private int hash(K key) - вычисляет хэш для ключа. Операция % table.length используется для умещения хэша в диапазон размера массива корзин.

public void put(K key, V value) - добавляет элемент в хэш-таблицу. Если в указанной корзине уже есть элемент с таким же ключом, то его значение обновляется.

public V get(K key) - возвращает значение по ключу. Если элемент с указанным ключом не найден, возвращается null.

public void remove(K key) - удаляет элемент из хэш-таблицы по ключу.

public int size() - возвращает текущий размер хэш-таблицы.

public boolean isEmpty() - проверяет, является ли хэш-таблица пустой

Этот код создает универсальную хэш-таблицу, которую можно использовать для хранения пар ключ-значение для любых типов данных.



Конструктор HashTable(int capacity):

Создается массив table корзин заданной емкости (capacity).

size инициализируется значением 0.

Метод hash(K key):

Вычисляет хэш для ключа (key) с использованием метода hashCode() объекта ключа.

Операция % table.length используется для умещения хэша в диапазон размера массива корзин.

Метод put(K key, V value):

Вычисляет хэш для ключа.

Если корзина (table[index]) для этого хэша пуста, создается новый связанный список (LinkedList).

Для каждого элемента в списке проверяется, есть ли элемент с таким же ключом. Если да, обновляется значение элемента.

Если элемент с таким ключом не найден, создается новый элемент Entry и добавляется в связанный список.

Увеличивается размер (size) таблицы.

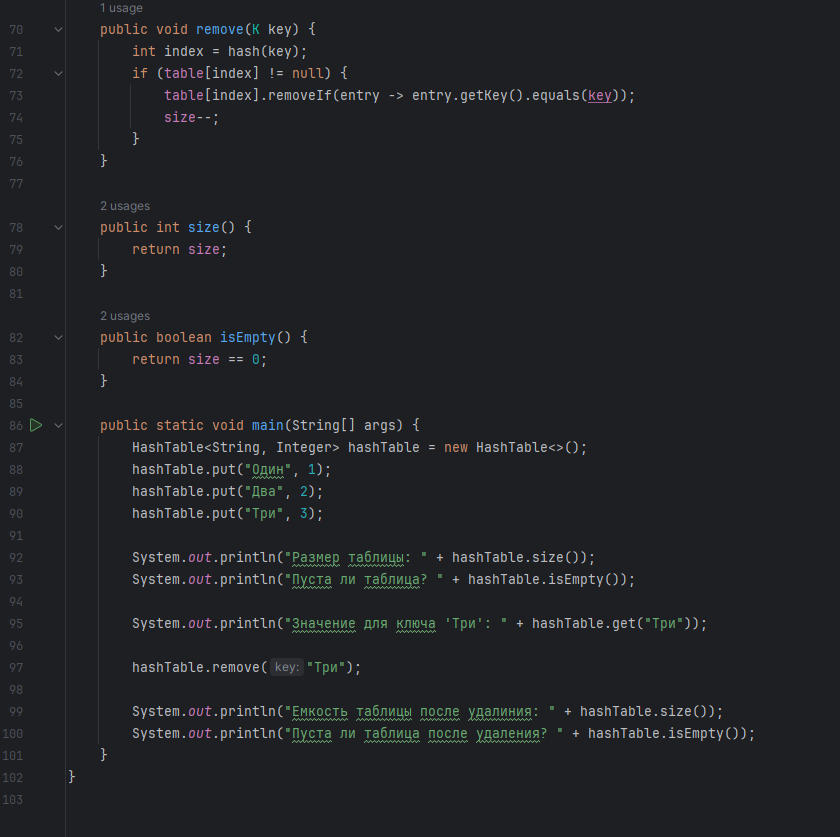
Метод get(K key):

Вычисляет хэш для ключа.

Если корзина для этого хэша не пуста, ищет элемент с заданным ключом в связанном списке.

Если элемент с таким ключом найден, возвращает его значение.

Если элемент не найден, возвращает null.



Метод remove(K key):

Вычисляет хэш для ключа.

Если корзина для этого хэша не пуста, удаляет все элементы в связанном списке с заданным ключом.

Уменьшается размер (size) таблицы на количество удаленных элементов.

Метод size():

Возвращает текущий размер таблицы.

Метод isEmpty():

Возвращает true, если таблица пуста (размер равен 0), в противном случае возвращает false.

Метод main(String[] args):

Создается экземпляр HashTable для строк в качестве ключей и целых чисел в качестве значений.

Добавляются несколько элементов с различными ключами и значениями.

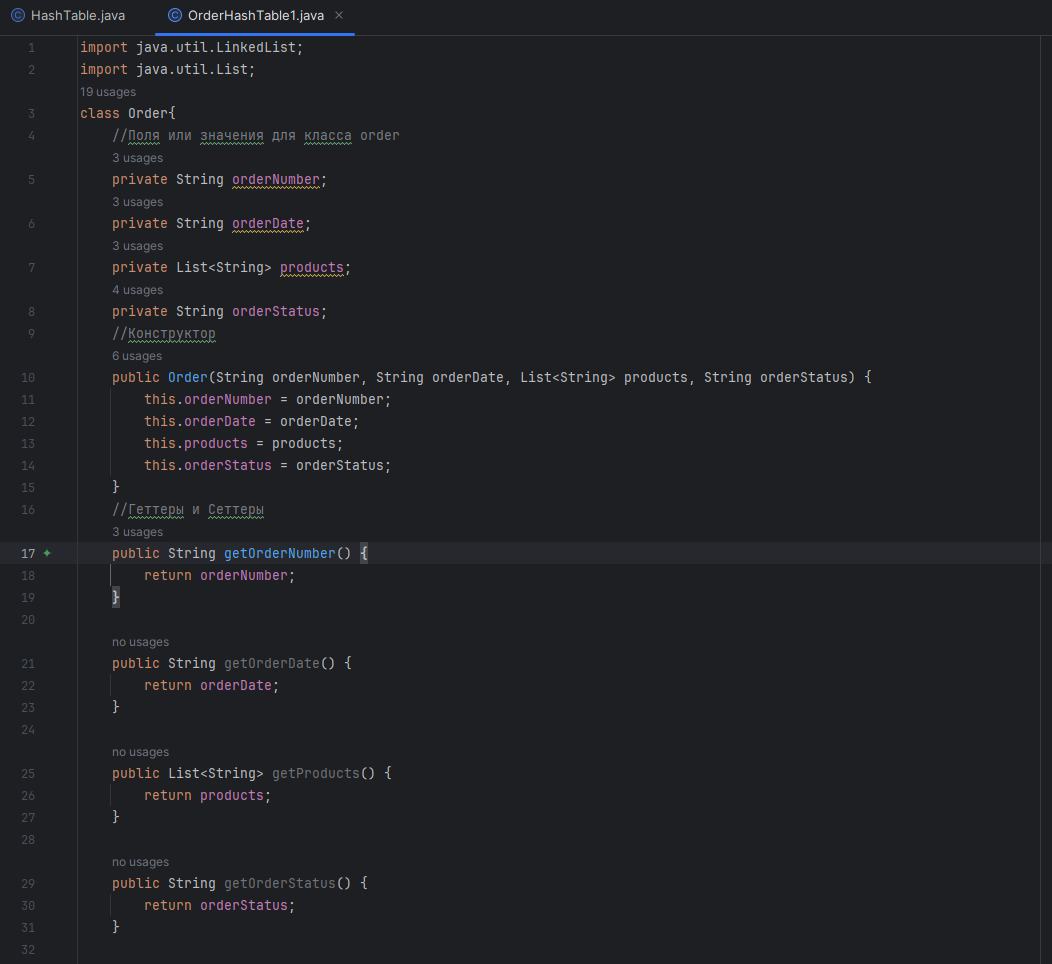
Выводится размер таблицы и проверяется, пуста ли таблица.

Выводится значение для ключа "Три".

Удаляется элемент с ключом "Три".

Проверяется размер таблицы после удаления и ее пустота.

***Задание 2***

******

Этот код представляет класс Order, предназначенный для представления информации о заказах. Вот пошаговое объяснение кода:

Поля класса Order:

orderNumber: Строковое значение, представляющее номер заказа.

orderDate: Строковое значение, представляющее дату заказа.

products: Список строк, представляющих продукты в заказе.

orderStatus: Строковое значение, представляющее статус заказа.

Конструктор Order:

Принимает значения для всех полей и устанавливает их в соответствии с переданными аргументами.

Геттеры:

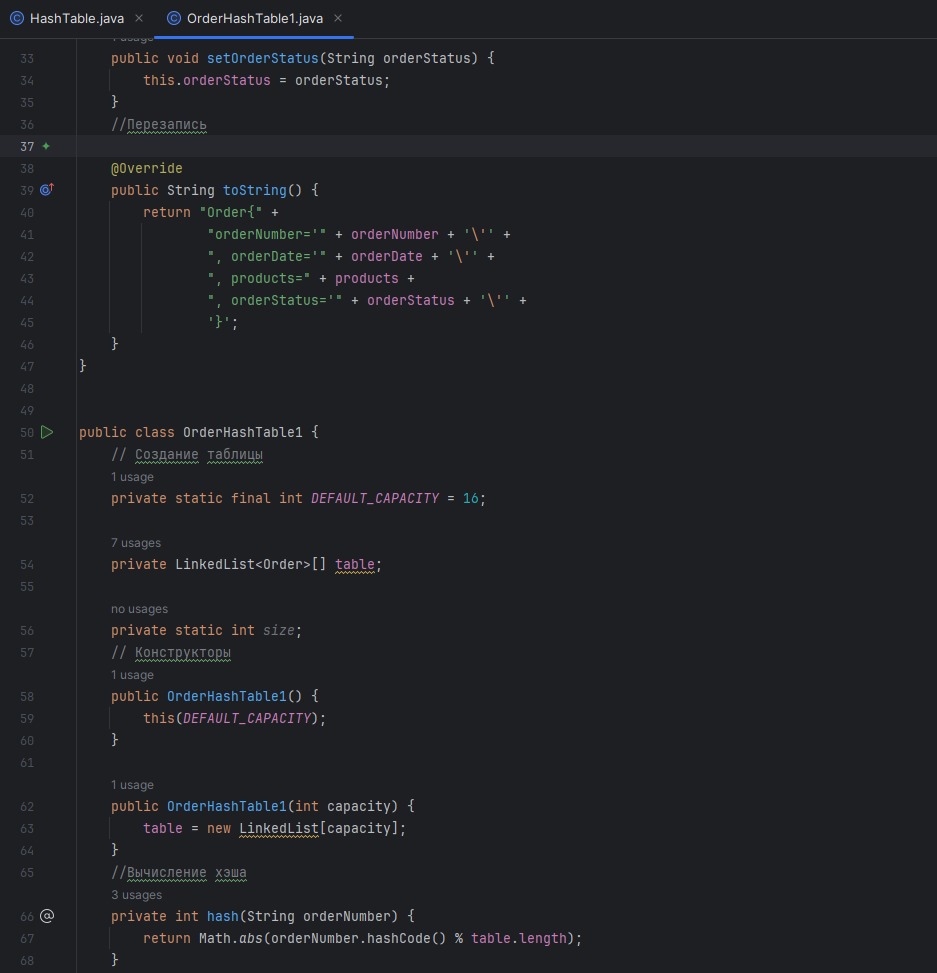
getOrderNumber(): Возвращает номер заказа.

getOrderDate(): Возвращает дату заказа.

getProducts(): Возвращает список продуктов в заказе.

getOrderStatus(): Возвращает статус заказа.

Класс Order предоставляет удобный способ создания объектов заказов, а также получения информации о них с использованием геттеров. Этот код представляет основу для работы с заказами в контексте системы управления заказами или подобных приложений.



Поле table:

Это массив связанных списков (LinkedList<Order>[]), предназначенный для хранения заказов. Каждый элемент массива может содержать связанный список заказов.

Поле size:

Статическое поле, которое, кажется, не используется в коде. Обычно, поле размера (size) используется для отслеживания количества элементов в хэш-таблице, но в данном случае, оно не участвует в логике хэш-таблицы.

Конструкторы:

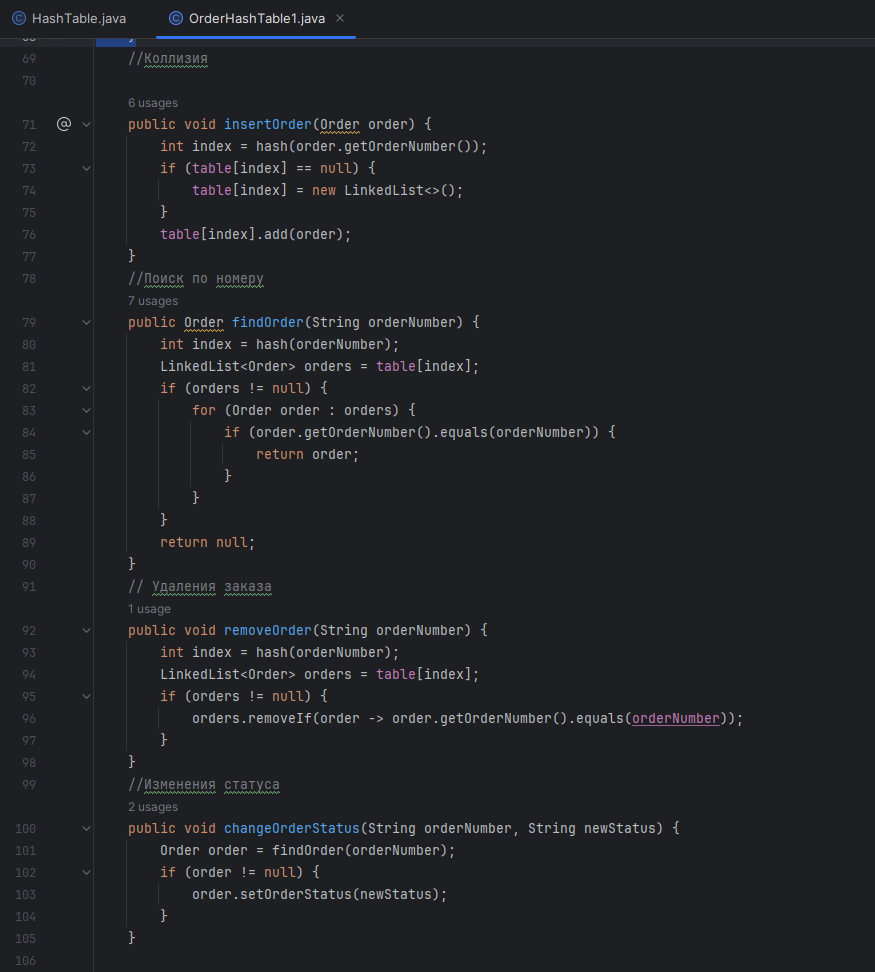
OrderHashTable1(): Конструктор по умолчанию, создает хэш-таблицу с размером по умолчанию (16).

OrderHashTable1(int capacity): Конструктор, позволяющий указать начальную емкость хэш-таблицы.

Метод hash:

Принимает номер заказа (orderNumber) и вычисляет хэш для его распределения по ячейкам массива. Используется операция % для получения остатка от деления хэш-кода номера заказа на текущий размер хэш-таблицы.

Важно отметить, что код предоставляет основную структуру хэш-таблицы, но в нем также присутствуют некоторые недочеты, такие как неудачное использование статического поля size и неполнота реализации методов, таких как вставка, поиск, удаление, изменение статуса заказа. Эти методы нуждаются в доработке, чтобы обеспечить правильное функционирование хэш-таблицы для управления заказами.



Метод insertOrder:

Принимает объект типа Order и вставляет его в хэш-таблицу.

Вычисляется хэш для номера заказа, получается индекс в массиве.

Если ячейка с соответствующим индексом пуста (равна null), создается новый связанный список (LinkedList<Order>), и заказ добавляется в этот список. Если ячейка не пуста, заказ просто добавляется в существующий список.

Метод findOrder:

Принимает номер заказа и возвращает объект Order, если заказ с таким номером найден, иначе возвращает null.

Вычисляется хэш для номера заказа, получается индекс в массиве.

Если ячейка с соответствующим индексом не пуста, происходит поиск заказа с нужным номером в связанном списке. Если заказ найден, он возвращается, в противном случае возвращается null.

Метод removeOrder:

Принимает номер заказа и удаляет заказ с этим номером из хэш-таблицы.

Вычисляется хэш для номера заказа, получается индекс в массиве.

Если ячейка с соответствующим индексом не пуста, удаляются все заказы с указанным номером из связанного списка.

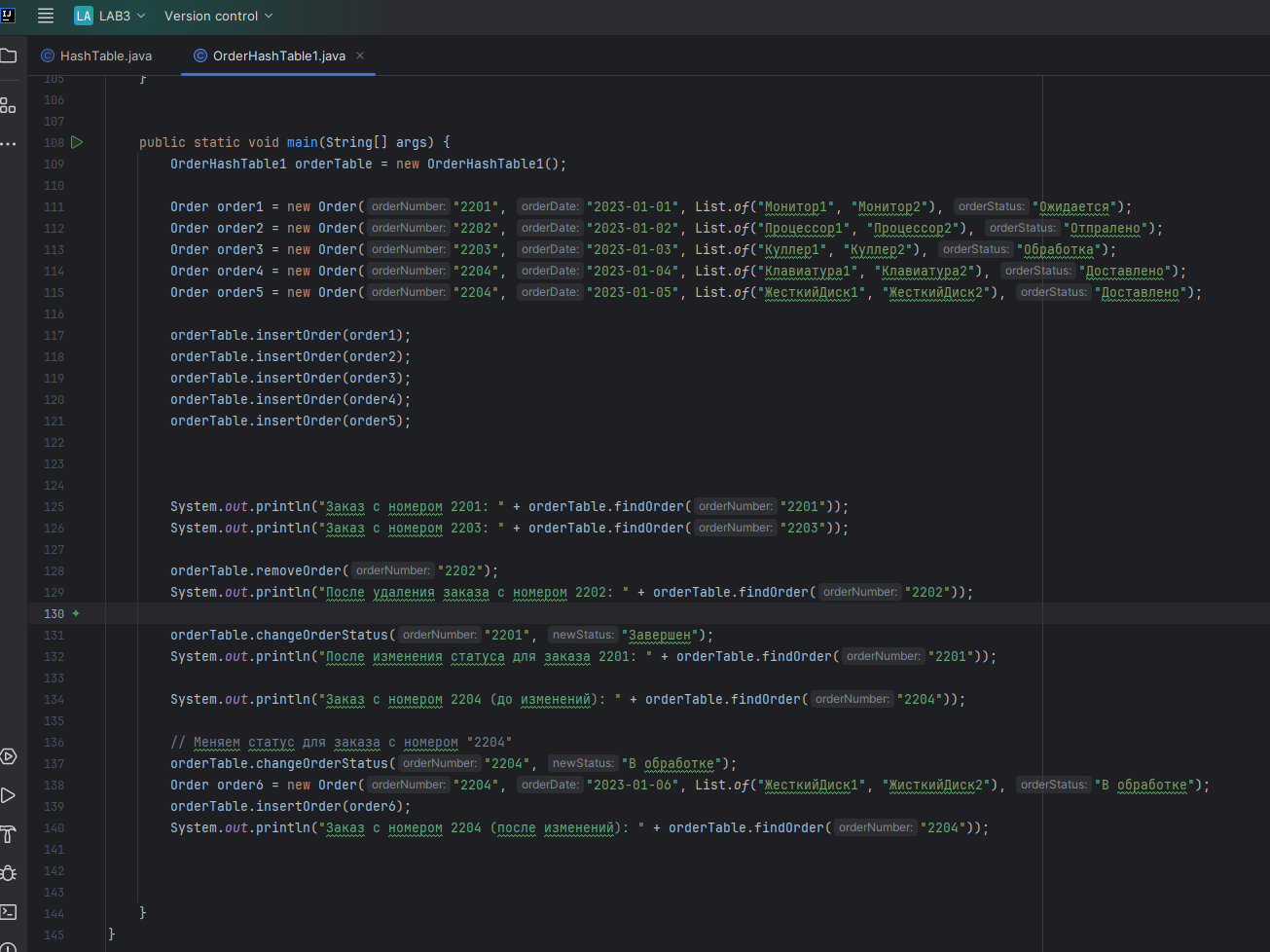
Метод changeOrderStatus:

Принимает номер заказа и новый статус заказа, изменяя статус заказа в соответствии с переданными значениями.

Использует метод findOrder для поиска заказа по номеру.

Если заказ найден, меняется его статус на переданный новый статус.

В целом, эти методы предоставляют базовый функционал для работы с хэш-таблицей заказов: вставка, поиск, удаление и изменение статуса заказа.



Создание экземпляра OrderHashTable1:

Создается объект OrderHashTable1, который представляет собой хэш-таблицу для хранения заказов.

Создание заказов и их добавление в хэш-таблицу:

Создаются пять объектов типа Order с разными номерами, датами, списками продуктов и статусами.

Заказы добавляются в хэш-таблицу с помощью метода insertOrder.

Вывод информации о заказах:

Выводится информация о заказах с номерами "2201" и "2203".

Удаление заказа:

Удаляется заказ с номером "2202", и выводится информация о заказе с этим номером (должен быть null, так как заказ удален).

Изменение статуса заказа:

Изменяется статус заказа с номером "2201" на "Завершен", и выводится информация о заказе с этим номером.

Изменение статуса и добавление нового заказа с одинаковым номером:

Изменяется статус заказа с номером "2204" на "В обработке".

Создается новый заказ с номером "2204", и его статус также устанавливается в "В обработке".

Выводится информация о заказе с номером "2204" после изменений.

https://github.com/G1QSTAR/Laba-itip.git