**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

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

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе №3.

Выполнил: студент группы \_\_\_БВТ2402\_\_

Стадник Андрей Романович

Москва, 2025

**Цель работы:** изучить и закрепить практические навыки работы с структурами данных для хранения пар «ключ-значение» в Java, освоить принципы хэширования, реализацию собственной хэш-таблицы с методом цепочек и использование встроенной коллекции HashMap, а также научиться эффективно управлять данными и выводить их в удобном формате.

**Задание 1.**

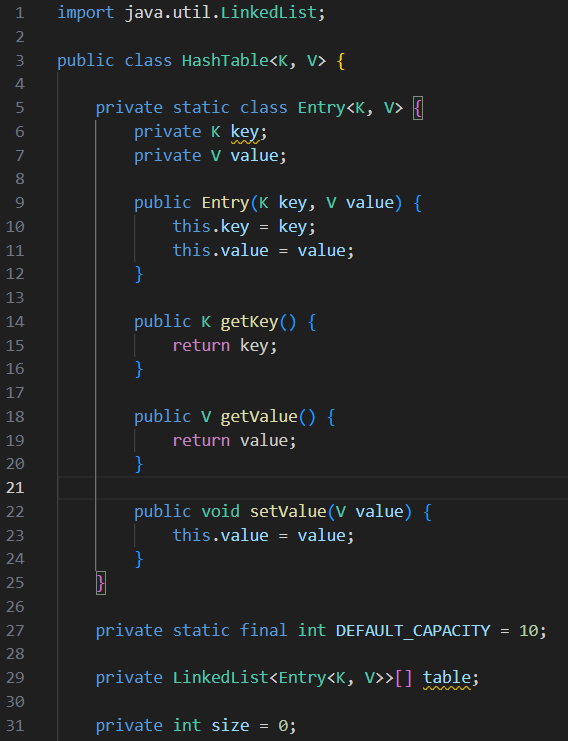
1. Создайте класс HashTable, который будет реализовывать хэш-таблицу с помощью метода цепочек.

2. Реализуйте методы put(key, value), get(key) и remove(key), которые добавляют, получают и удаляют пары «ключ-значение» соответственно.

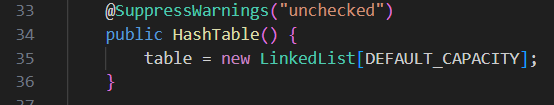
3. Добавьте методы size() и isEmpty(), которые возвращают количество элементов в таблице и проверяют, пуста ли она.

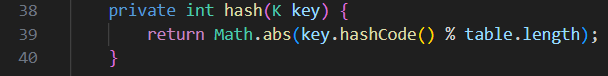
**Решение:**

1) Сначала был создан класс HashTable, внутри которого определён вложенный статический класс Entry. Класс Entry хранит ключ и значение и предоставляет методы для получения ключа и значения и для изменения значения.

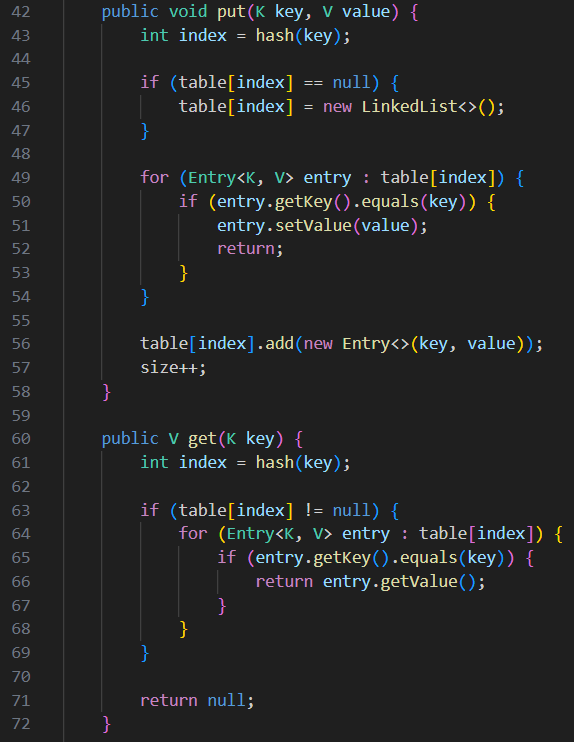
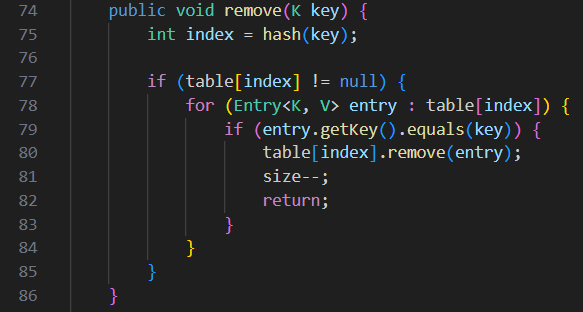


2) Для хранения элементов был создан массив корзин LinkedList<Entry<K, V>>[] table. Каждая корзина представляет собой связанный список. Массив инициализируется в конструкторе класса, его размер задаётся константой DEFAULT\_CAPACITY.

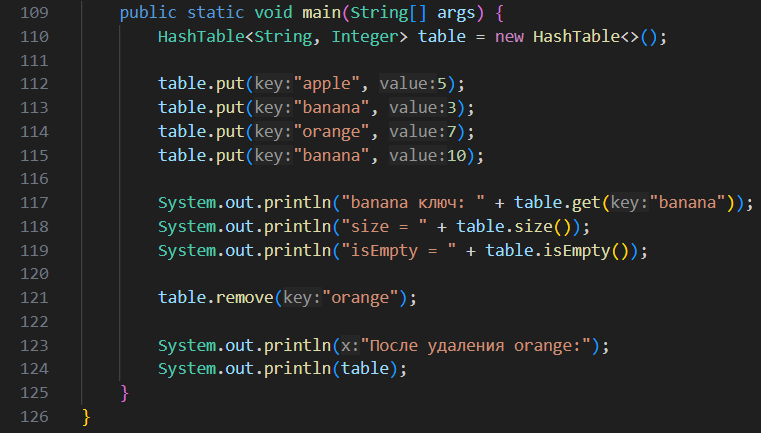


3) Хэш-функция hash(K key) вычисляет индекс корзины по ключу с использованием метода hashCode() и оператора остатка от деления. Результат приводится к положительному числу. 

4) Метод put(K key, V value) добавляет новый элемент или обновляет значение существующего. Сначала вычисляется индекс корзины, при необходимости создаётся новый список, затем проверяется наличие ключа в корзине. Если ключ уже присутствует, обновляется значение, иначе создаётся новый объект Entry и добавляется в список. Методы get(K key) и remove(K key) находят и удаляют элементы по ключу, проходя по всем элементам корзины.



5) Реализованы методы size() и isEmpty(). Переопределён метод toString(), который выводит все пары «ключ-значение» с помощью StringBuilder. 

6) В методе main создавался объект хэш-таблицы, добавлялись элементы, проверялась возможность обновления значений, получение и удаление элементов, а также вывод всей таблицы на экран. 

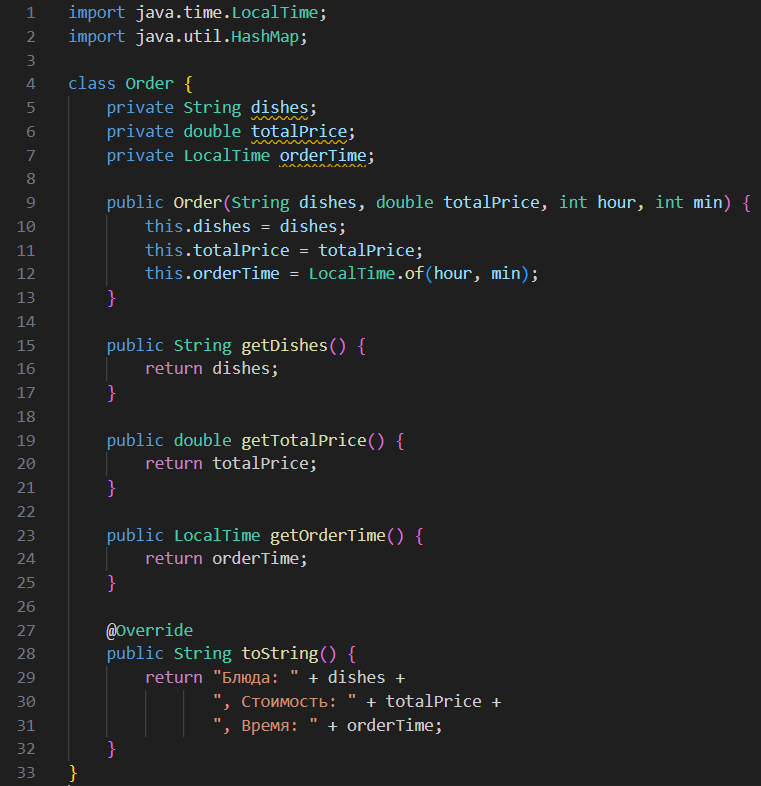
**Задание 2. Работа с встроенным классом HashMap.**

**10 Вариант:**

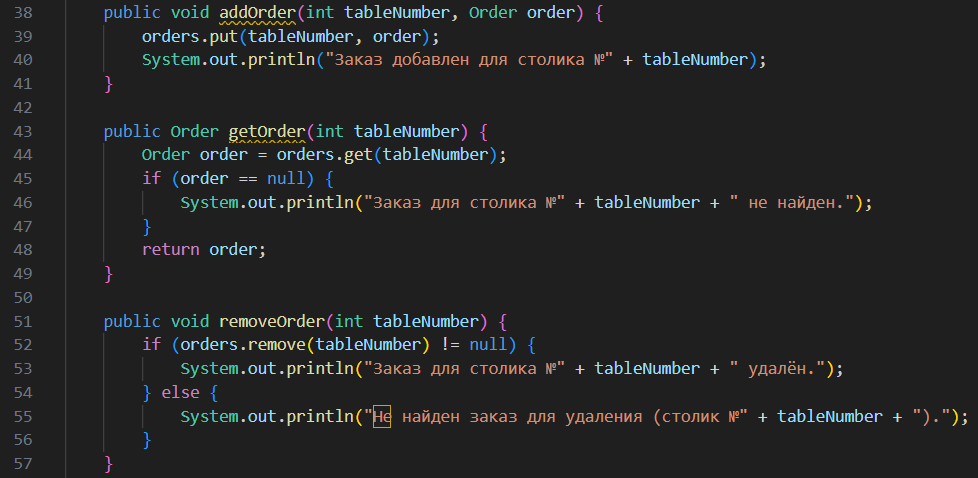
Реализация хэш-таблицы для хранения данных о заказах в ресторане. Ключом будет номер столика или заказа, а значением — объект класса Order, содержащий информацию о блюдах, стоимости и времени заказа. Необходимо реализовать операции вставки, поиска и удаления заказа по номеру столика.

**Решение:**

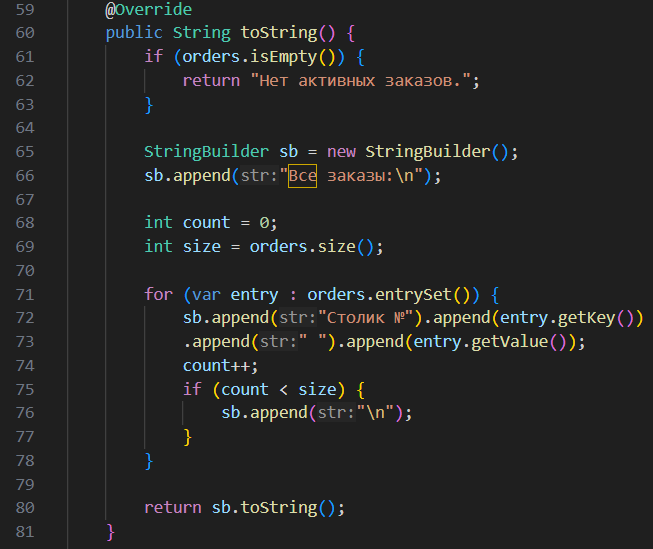
1) Сначала был создан класс Order, который хранит информацию о заказе: блюда, стоимость и время заказа с использованием LocalTime. Переопределён метод toString() для вывода информации о заказе.

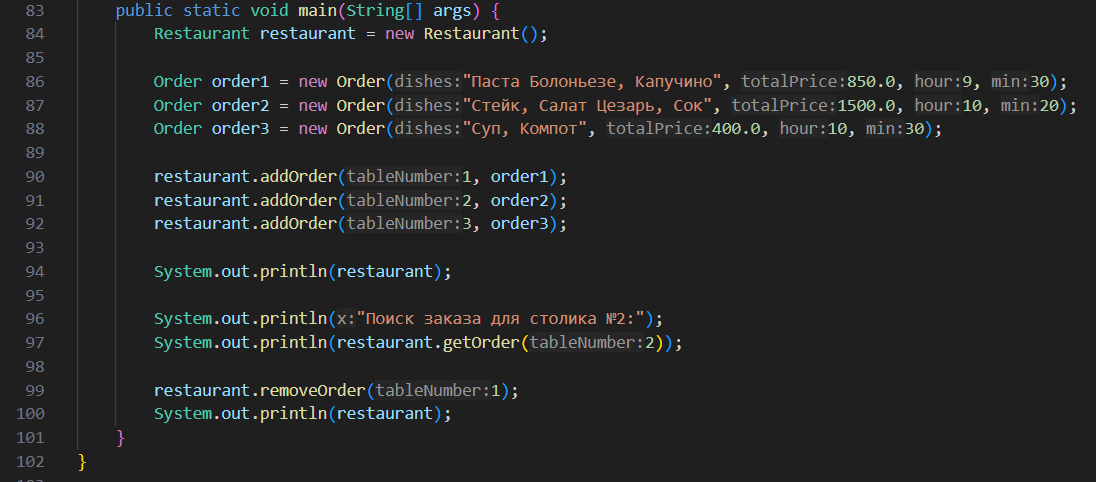


2) Затем создавался класс Restaurant, который содержит HashMap<Integer, Order> для хранения заказов. Ключ — номер столика, значение — объект Order. 

3) Метод addOrder добавляет заказ в HashMap по номеру столика и выводит сообщение о добавлении. Метод getOrder ищет заказ по номеру столика и возвращает объект Order. Если заказ не найден, выводится соответствующее сообщение. Метод removeOrder удаляет заказ по номеру столика. Если заказ отсутствует, выводится сообщение. 

4) Переопределение метода toString() в классе Restaurant формирует строку со всеми активными заказами, перебирая пары ключ-значение через entrySet(). Перенос строки добавляется только между элементами.



5) В методе main создавался объект ресторана и несколько заказов, которые добавлялись в систему. Проверялись операции добавления, поиска, удаления и вывода всех заказов. 

**Вывод:** В ходе работы были изучены и закреплены практические навыки работы с структурами данных для хранения пар «ключ-значение» в Java. Были реализованы собственная хэш-таблица с методом цепочек и программа для управления заказами с использованием HashMap. Отработаны операции добавления, получения, удаления и вывода данных, а также перебор пар ключ-значение через entrySet(). Все методы протестированы через main, что подтвердило корректную работу реализованных структур и программ.

**Ссылка на GitHub:** <https://github.com/pmr-fiz/Laboratory-2-course/tree/main/Laboratory%203>