**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 «Математической кибернетики и информационных технологий»

Информационные технологии и программирование

Лабораторная работа №3:

Выполнение заданий с хэш-таблицей

Выполнила:

студентка группы БВТ2203

Залялиева Ильвина

Москва 2023

**Цель работы:** решение заданий.

**Задание:** 1) Создайте класс HashTable, который будет реализовывать хэштаблицу с помощью метода цепочек. 2) Реализация хэш-таблицы для хранения информации о сотрудниках в компании. Ключом будет идентификационный номер сотрудника, а значением - объект класса Employee, содержащий данные о имени, должности и заработной плате. Необходимо реализовать операции вставки, поиска и удаления сотрудника по ID.

**Ход работы**

1. Сначала импортирую библиотеку, которая будет использоваться для хранения цепочек в хэш-таблице. Далее начинается определение класса HashTable, который параметризован двумя типами ключ и значение. Cчетчик для размера таблицы. Затем идет массив связанных списков, который представляет основную структуру хэш-таблицы. public HashTable() конструктор класса HashTable, который инициализирует массив table c некой длиной.

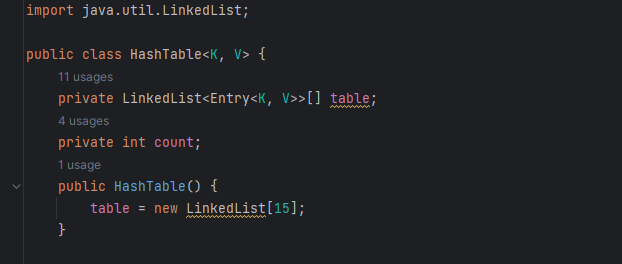


Рисунок 1 – основные моменты.

1. После пишу метод put, который добавляет пару "ключ-значение" в хэш-таблицу. Он вычисляет хеш ключа, находит соответствующее значение и добавляет элемент в список.

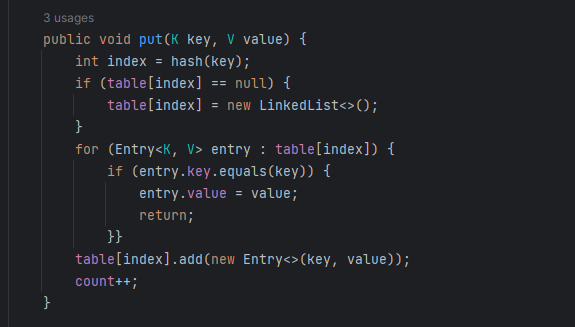


Рисунок 2 – метод put.

1. Этот метод получает значение, связанное с заданным ключом, вычисляет хеш ключа, находит соответствующее значение и ищет элемент в списке.

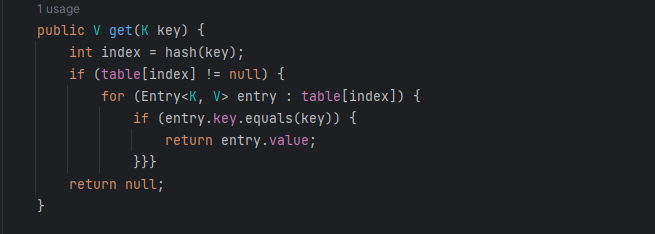


Рисунок 3 – метод get.

1. Следующий метод удаляет пару "ключ-значение" по заданному ключу. Size возвращает количество элементов в хэш-таблице, обращаясь к переменной count.

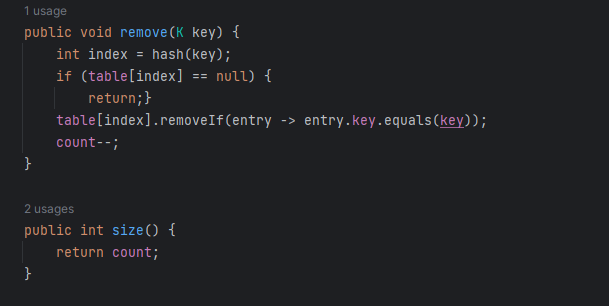


Рисунок 4 – методы удаления, размера.

1. isEmpty() проверяет, пуста ли хэш-таблица по переменной. hash() - приватный метод, вычисляет хеш ключа с использованием метода hashCode() и применяет операцию остатка для получения индекса в массиве table. Entry<K, V> приватный статический класс, представляющий элемент "ключ-значение", который содержит поля для ключа и значения. Лучше писать в начале.

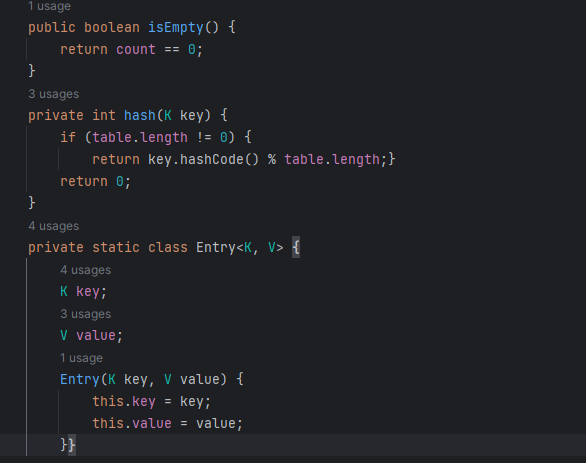
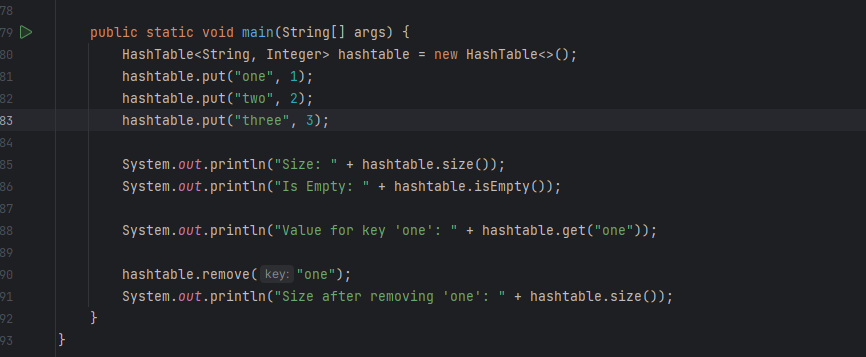


Рисунок 5 – проверка на наличие, вычисление хэша.

1. Вывод характеристик хэш таблицы.



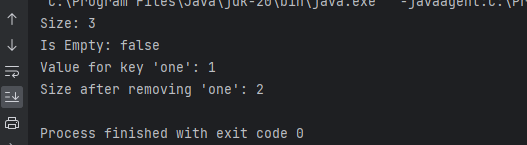
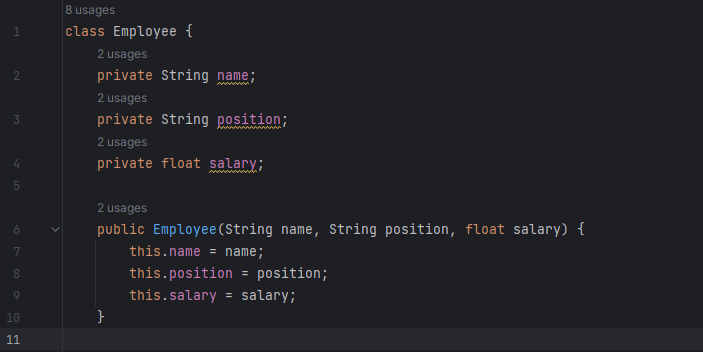


Рисунок 6, 7 – результат.

1. Задание 2. Первые строчки кода определяют класс Employee. Он имеет три члена данных, которые используются для хранения информации о сотруднике: name, position и salary. Они являются приватными , могут быть доступны только внутри класса Employee. Далее определение конструктора класса. Это позволяет установить начальные значения для Employee при его создании. После прописываю методы доступа (геттеры) для получения значений членов данных класса.



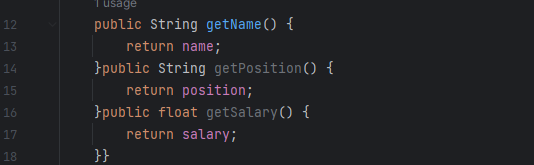


Рисунок 7,8 – класс Employee.

1. Здесь я импортирую класс HashMap, потом определяю класс EmployeeHashTable, который содержит приватное поле - хеш-таблица, где ключами являются ID сотрудника, а значениями - объекты класса Employee.



Рисунок 9 – хэш-таблица.

1. После идут методы addEmployee, getEmployee и removeEmployee, которые позволяют добавлять, получать и удалять сотрудников из хеш-таблицы . Метод addEmployee добавляет нового сотрудника в таблицу с использованием предоставленного ID в качестве ключа, аналогично с остальными.

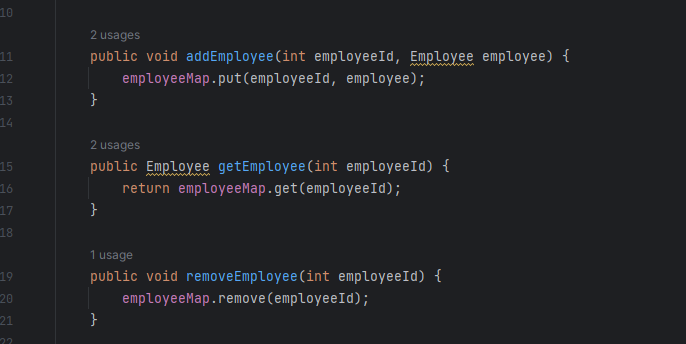


Рисунок 10 – методы.

1. Пишу вывод и результат.



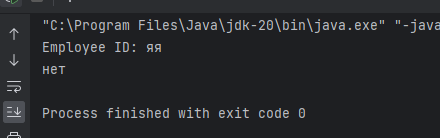


Рисунок 11, 12 – вывод.

Вывод: в ходе лабораторной работы я выполнила задания, связанные с хэш-таблицей.