**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

Работа с коллекциями

Выполнил: Студент группы

БВТ2303

Кунецкий Владислав

Москва

2024

Цели работы:

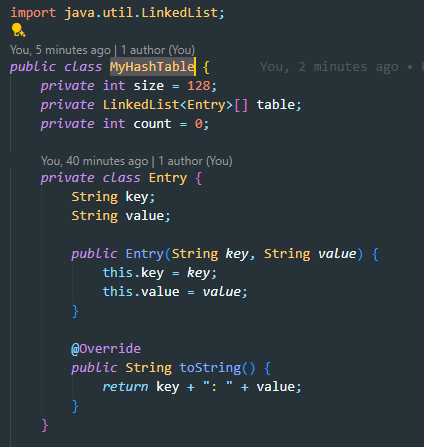
- Познакомиться с коллекциями, представленными в языке Java.

- Реализовать свой аналог HashTable

- Поработать со встроенной HashMap.

-Ход работы:

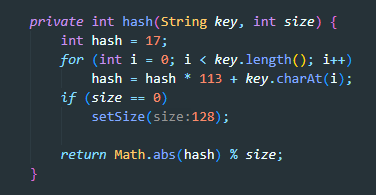
После изучения принципов хеш-таблиц я приступил к реализации своей. Для начала я создал основной класс MyHashTable. Я решил использовать массив связанных списков, в которые будут помещаться объекты, содержащие ключ-значение. Массив будет динамическим, изменение размера будет производиться в зависимости от количества элементов.



Скрин 1 – Поля класса

Поле size - начальный размер массива, массив списков это собственно и есть наша основная структура данных. Count будет отслеживать количество элементов внутри нашей мапы, а класс Enty описывает объекты, которые мы будет помещать внутрь.

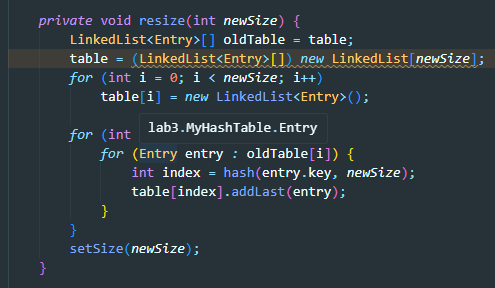
Далее я написал свою хеш-функцию, преобразующую ключ в индекс массива.



Скрин 2 – хеш-функция

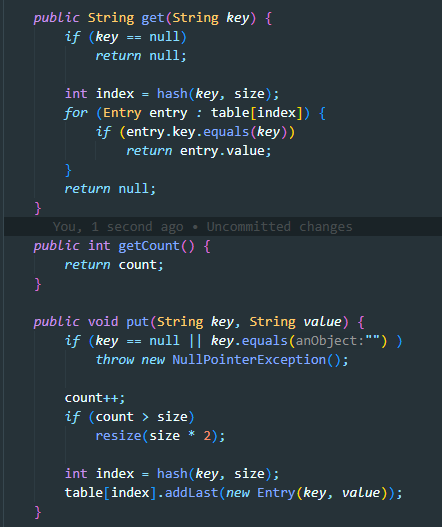
Для лучшего распределения использовал простые числа. Сдвиги, хоть и являются довольно быстрыми, но обеспечивают плохую распределяемость, поэтому от них я отказался.

Важным этапом стало написание метода, для изменения длинны массива. В нем мы производим копирование старого массива и создаем новый, заменяющий старый. После в цикле копируем значения, пересчитывая их индекс.



Скрин 3 – Метод переопределения длинны массива

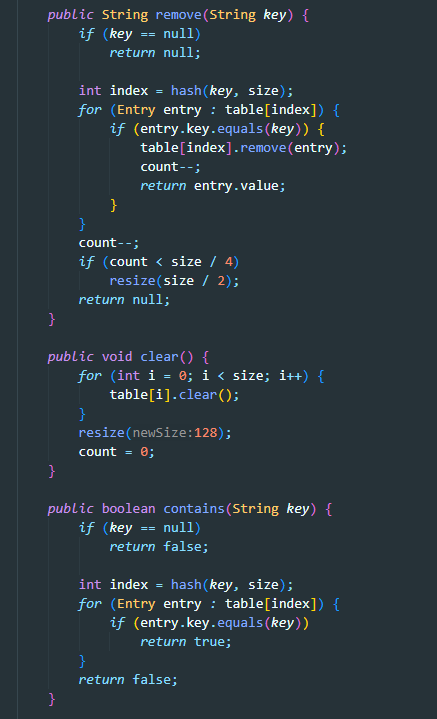
После реализовал основную логику.



Скрин 4 – Вставка и получение значий.

При вставке мы генерируем индекс элемента с помощью хеш-функции и производим вставку объекта в конце связанного списка по индексу. В случае, когда количество элементов превышает размер, увеличиваем его.

В гетер мы получаем также хеш, а далее проходимся циклом по элементам списка и ищем нужный ключ.



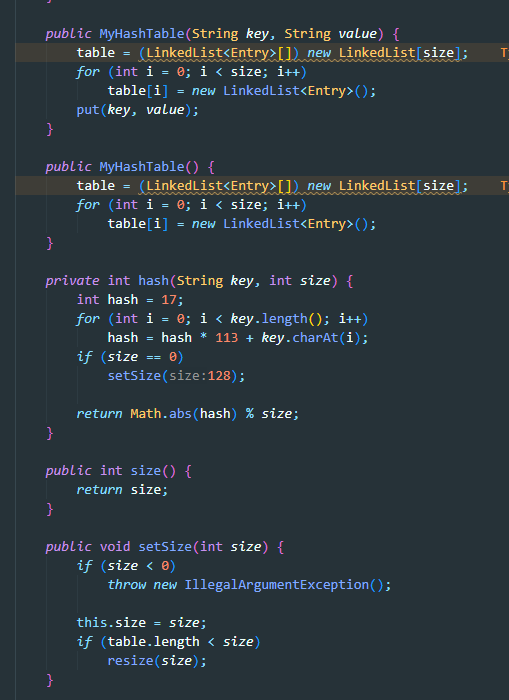
Скрин 5 – Удаление и проверка наличия.

Далее я реализовал удаление, очистку и проверку на наличие(скрин 5). Логика везде одинаковая

Вспомогательные методы



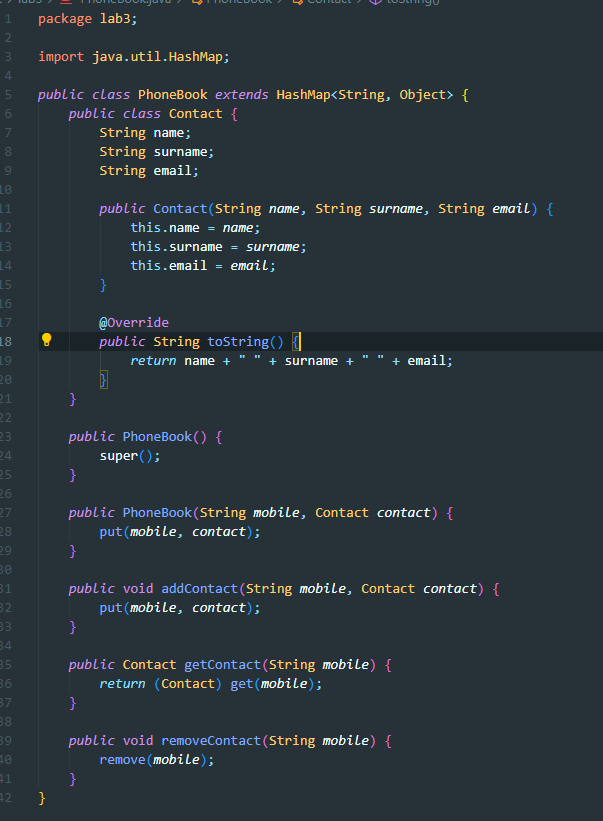
Скрин 6 – показ содержимого таблицы



Скрин 7 - Конструкторы и методы для изменения/получения размера

Во второй части работы мне предстояло написать хэш-таблицу для хранения контактов в телефонной книге.

Я наследовался от класса HashMap, определил объект Contact и определил основные методы.



Скрин 8 – Вторая часть работы