Министерство цифрового развития, связи и массовых коммуникаций

Российской Федерации

Ордена Трудового Красного Знамени

Федерально государственное бюджетное образовательное учреждение

высшего образования

МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ СВЯЗИ И ИНФОРМАТИКИ

Кафедра «Математической кибернетики и информационных технологий»

Лабораторная работа №3

По дисциплине

«Информационные технологии и программирование»

Выполнил:

Студент группы БВТ2203

Бородин К.Н.

Москва 2023

ВВЕДЕНИЕ

Цель работы: реализовать хеш-таблицу на ЯП Java.

Задания:

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

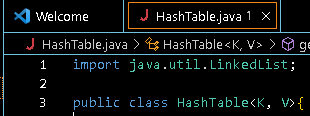
2. Реализуйте методы put(key, value), get(key) и remove(key), которые добавляют, получают и удаляют пары «ключ-значение» соответственно.

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

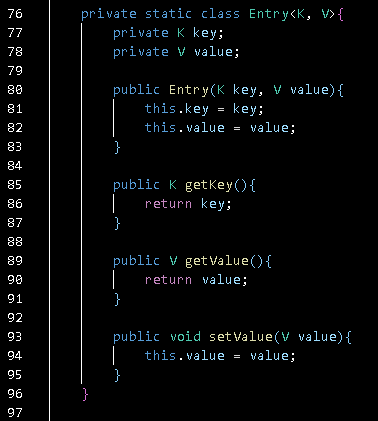
4. Реализация хэш-таблицы для хранения информации о книгах в библиотеке. Ключом будет ISBN книги, а значением - объект класса Book, содержащий информацию о названии, авторе и количестве копий. Необходимо реализовать операции вставки, поиска и удаления книги по ISBN.

ХОД РАБОТЫ.

Импортирую библиотеку для создания связанных списков, а также создаём класса хештаблицы с возможностью использования различных типов данных(generic class), рисунок 1.

  
Рисунок 1 – импорт библиотеки и создание класса.

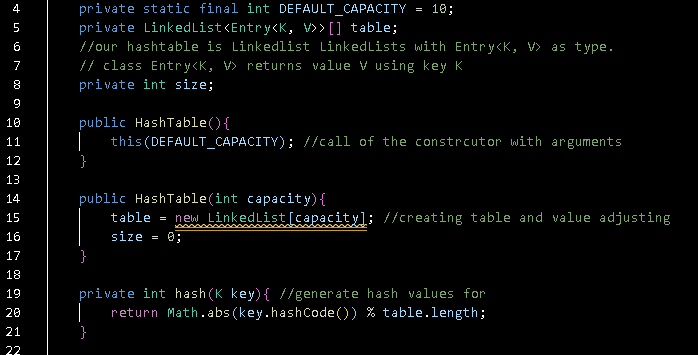
Создаю вложенный класс, с помощью которого можно связывать ключ и значение. В нём есть конструктор, геттеры для ключа и значения, сеттер для значения, рисунок 2.

  
Рисунок 2 – создание вложенного класса.

Создаю неизменяемую переменную для хранения максимальной вместимости хештаблицы по умолчанию. Создаю переменную, в которой будет храниться сама таблица, с которой мы будем работать. Это будет связанный список связанных списков типа созданного раннее Entry. Создаю переменную для хранения размера.

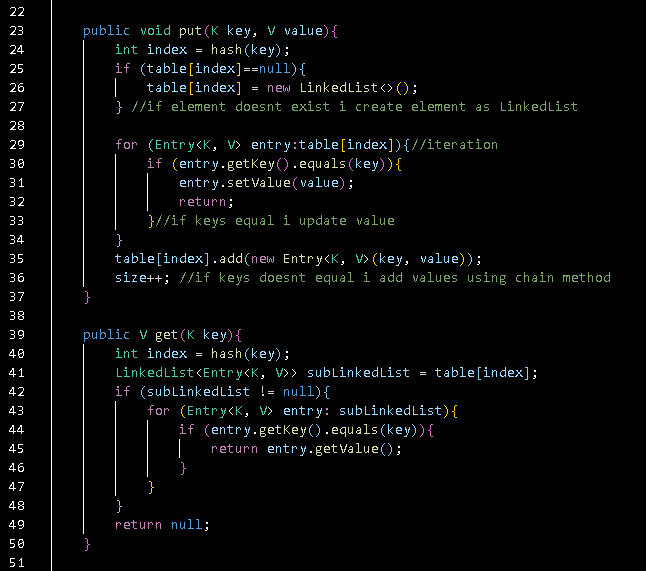
Создаю конструктор по умолчанию, который будет вызывать параметрический конструктор, который будет создавать таблицу из связанных списков и размер этой таблицы.

Создаю метод для хеширования значения. Он будет брать модуль от хешкода ключа, и от этого значения брать остаток от деления на длину таблицы, чтобы не выходило число, большее вместимости таблицы, рисунок 3.

  
Рисунок 3 – создание полей, конструкторов и методов.

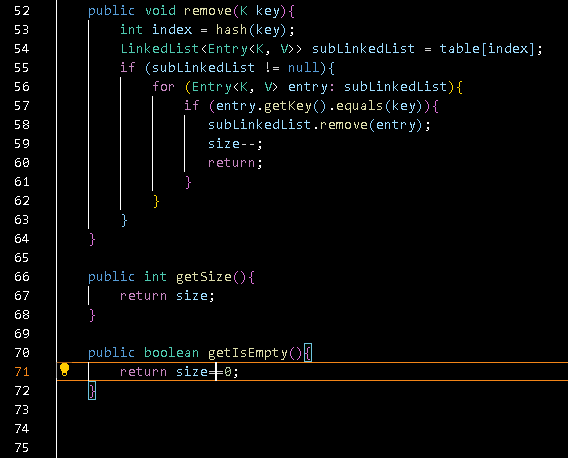
Метод для добавления элемента по ключу и значению. По ключу получаю его хеш. Если в нашей таблице по хешу нет элемента, то мы создаём по этому хешу связанный список. Затем итерируюсь по элементам связанного списка по хешу. Если по ключу есть значение, то я его заменяю и завершаю выполнение метода. Если мы изменили элемент, то метод выполняется дальше. Добавляю по хешу в связанный индекс элемент и обновляю размер таблицы.

Метод получение значения по ключу. Получаю индекс по хешу ключа. Затем сохраняю в переменную связанный список по индексу. В этом связанном списке ищу элемент по ключу. Если он найдет, то возвращаю значение, если нет, то возвращаю null, рисунок 4.

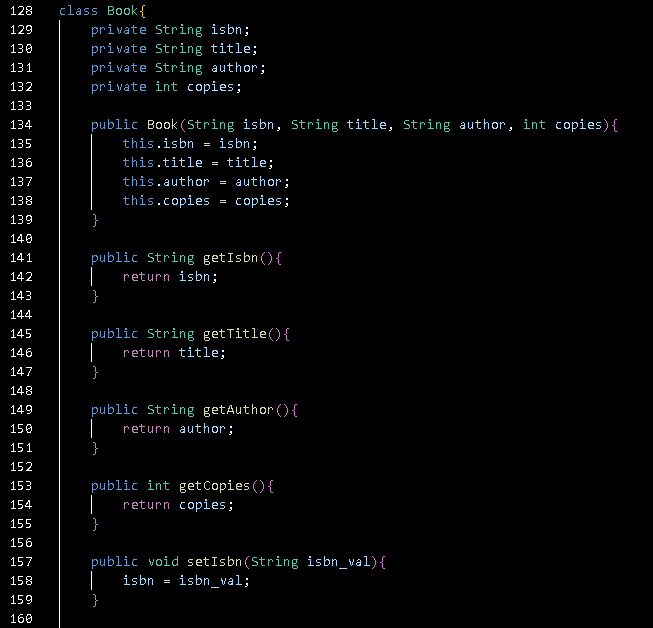
  
Рисунок 4 – методы put и get.

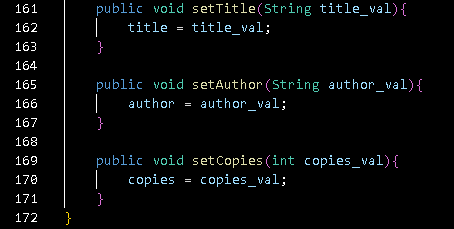
Создаю метод для удаления значения по его. Метод аналогичен методу get, за исключением тела цикла. В нём я удаляю элемент и обновляю значение размера таблицы.

Создаю геттеры для получения размера таблицы и проверка на пустоту, рисунок 5.

  
Рисунок 5 – создание методов remove, gitSize, getIsEmpty.

Создаю класс для книги. В нём будут поля для isbn, автора, названия и количества копий, конструкторы параметрический и по умолчанию, геттеры и сеттеры для переменных, рисунки 6-7.

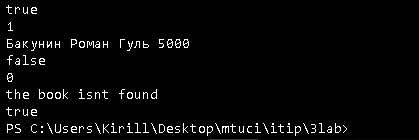
  
Рисунок 6 – создание класса для книг.

  
Рисунок 7 – создание класса для книг.

Создаю метод, где буду проверять работоспособность класса. В нём создаю экзмемпляр класса и проверяю его на пустоту. Затем создаю класс книги и вставляю его в хештаблицу. Затем получаю значение элемента по его ключу. Затем снова проверяю на пустоту. После удаляю этот элемент, получаю значение по ключу и проверяю на пустоту, рисунок 8.

  
Рисунок 8 – метод проверки таблицы.

Вывод программы представлен на рисунке 9.

  
Рисунок 9 – вывод программы.