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»

Лабораторная работа №3

**по дисциплине**

**«Информационные технологии и программирование»**

Выполнил: студент гр. БВТ2201

Аблязов И.В.

Проверил:

Москва, 2023 г.

# ВВЕДЕНИЕ

Цель задания: создать класс HashTable, который будет реализовывать хэш-таблицу с помощью метода цепочек. С помощью данного класса реализовать хэш-таблицу для хранения ифнормации о заказах в интернет-магазине.

В классе необходимо реализовать методы:

* put(key, value)
* get(key)
* remove(key)
* size()
* isEmpty()

При создании хэш-таблицы должны быть выполнены следующие условия:

* Ключом является номер заказа, а значением – объект класса Order, содержащий поля дата заказа, список товаров и статус заказа.
* Необходимо реализовать операции вставки, поиска и удаления заказа по номеру
* Необходимо реализовать метод для изменения статуса заказа

# ОСНОВНАЯ ЧАСТЬ

## Задание 1

Для начала был создан класс Entry. Это класс, представляющий единичную пару ключ-значение. Он имеет всего 2 поля, конструктор, конструктор по умолчанию, 2 геттера и 1 сеттер. Полный код данного класса показан на рисунке 1.

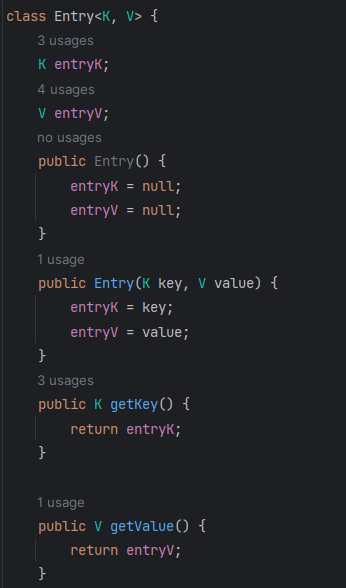


Рисунок 1 Класс Entry

После этого был создан необходимый класс HashTable. Он имеет 1 поле, 1 статическую переменную, конструкторы и 6 методов. Изначальная хэш-таблица будет иметь всего 10 индексов. Выбор индекса реализован при помощи метода hash.

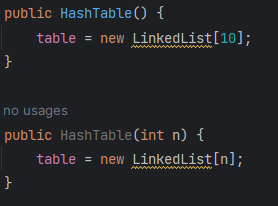


Рисунок 2 Конструкторы

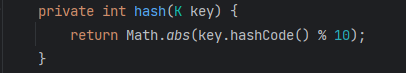


Рисунок 3 Hash метод

2 других метода работают со статической переменной size. Первый метод возвращает размер хэш-таблицы, а второй – boolean значение (True, если таблица пуста, и False, если таблица имеет какой-то размер).

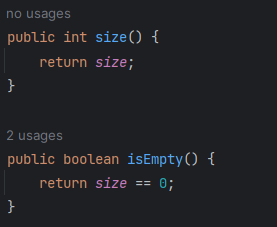


Рисунок 4 Методы size() и isEmpty()

В методе put(key, value) мы обращаемся к индексу, который создается для очередного ключа. Если до этого по данному индексу не было сохранено других пар ключ-значение, мы создаем новый список. После чего проверяем, встречался ли данный ключ до этого. Если данный ключ уже существует в таблице, то мы перезаписываем его значение. Если же до этого не был объявлен такой ключ, таблица пополняется новой парой ключ-значение, а ее размер увеличивается на 1.

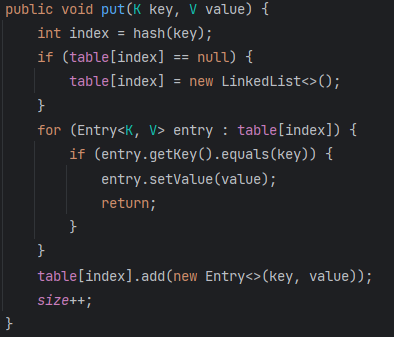


Рисунок 5 Метод put(key, value)

В методе get(key) мы проверяем существование ключа, после чего выводим его значение.

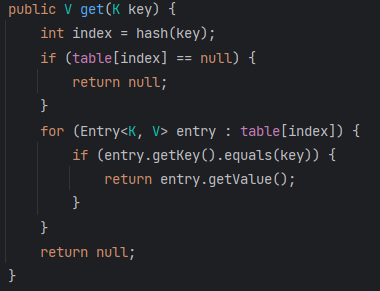


Рисунок 6 Метод get(key)

В методе remove(key) мы проверяем существование ключа, после чего удаляем его пару ключ-значение и уменьшаем размер таблицы на 1.

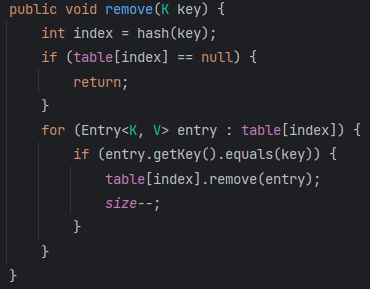


Рисунок 7 Метод remove(key)

Теперь проверим всё созданное ранее. Код проверки и результат проверки указаны ниже:

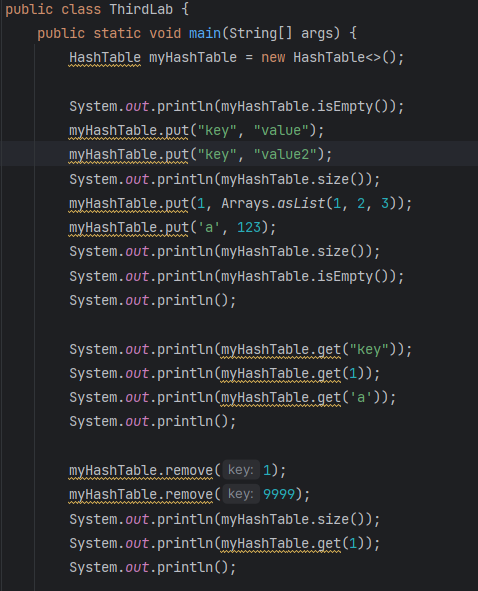


Рисунок 8 Проверка работы класса

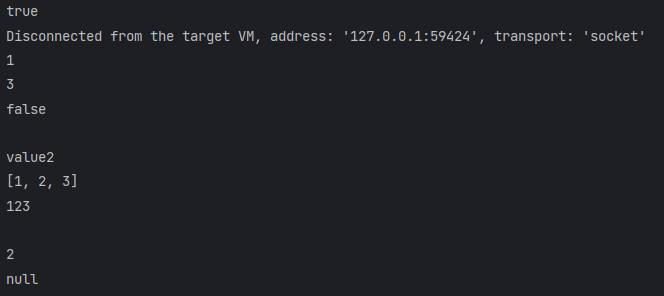


Рисунок 9 Результат работы программы

## Задание 2

Для выполнения этого задания был реализован класс Order, состоящий из полей orderDate, orderList и orderStatus, конструкторов, геттеров и сеттеров. Полный код данного метода показан на рисунках 10 и 11.

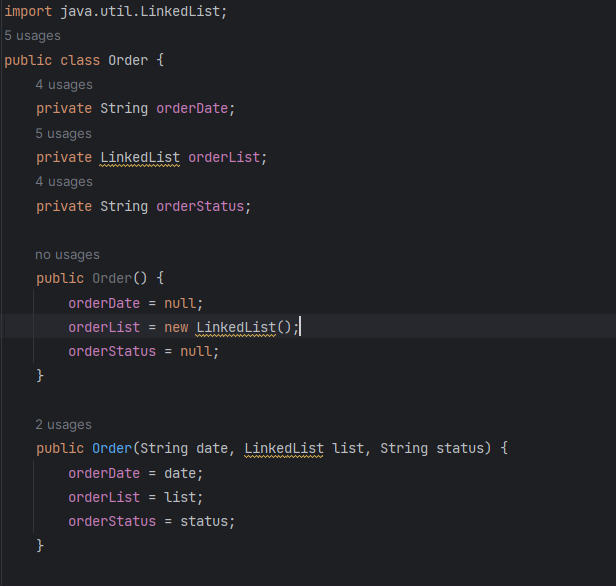


Рисунок 10 Класс Order (1)

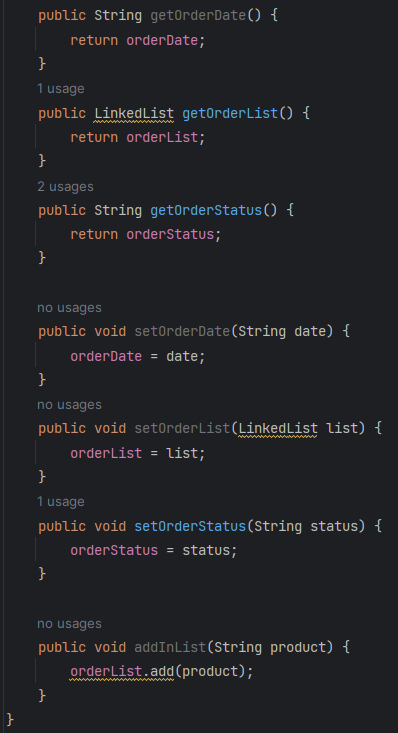


Рисунок 11 Класс Order (2)

После чего была создана необходимая таблица, где заранее указывается, что ключами являются целые числа, а значениями – объекты типа Order.



Рисунок 12 Задание 2

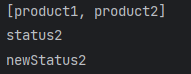


Рисунок 13 Результат работы программы

# ВЫВОД

Данная лабораторная работа помогла лучше освоить язык Java. Была проделана работа с классами и дженериками. Все знания были закреплены на практике и сохранены на GitHub: <https://github.com/HungryM1NT/ITIP>