Санкт-Петербургский политехнический университет Петра Великого

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**Отчет по курсовому проекту**

Хеш-таблица с двойным хешированием

Дисциплина: «Алгоритмы и структуры данных»

Выполнил студент гр. 3530901/00003 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кузнецов К. С.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ахин М. Х.

(подпись)

“ ” 2022 г.

Санкт-Петербург

2022

Содержание

[Техническое задание 3](#_Toc102078454)

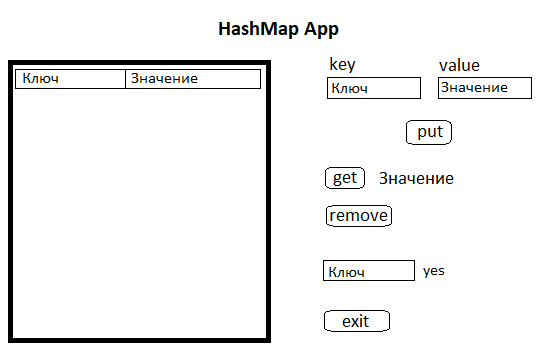
[Метод решения 5](#_Toc102078455)

[Листинг программы 6](#_Toc102078456)

[Работа программы 7](#_Toc102078457)

# Техническое задание

Реализация в виде класса на Java одного из видов хеш-таблицы. Класс таблицы должен реализовывать соответствующий интерфейс Java в обобщённом виде (Set или Map). Класс должен обеспечивать возможность неограниченного роста размера таблицы (в пределах памяти компьютера, естественно). Класс может обеспечивать или нет предсказуемый порядок перебора элементов (предсказуемый порядок, естественно, сложнее). Хеш-таблица с двойным хешированием.

* GUI приложение на JavaFX
* 
* При нажатии на кнопку «put» добавляем пару (ключ, значение).  
  При нажатии на кнопку «get» получаем значение к выбранному ключу.  
  При нажатии на кнопку «remove» удаляем выбранную пару.
* Исходные данные вводятся в текстовые поля «key» и «value».
* В списке слева от кнопок появляется добавленный ключ, выбрав который и нажав на кнопку «get», можно получить значение. Вводом ключа в текстовое поле, находящееся справа под копками, можно проверить его вхождение. Выход из приложения осуществляется кнопкой «exit».
* В программе будет использоваться двойное хеширование.
* «GitHub» репозиторий: <https://github.com/qkonstantin/DoubleHashtableGUI>

# Метод решения

Хеш-таблица — это контейнер, который используют, если хотят быстро выполнять операции вставки/удаления/нахождения. Мы определяем функцию хеширования, которая по каждому входящему элементу будет определять натуральное число. А уже дальше по этому натуральному числу мы будем класть элемент в (допустим) массив. Тогда, имея такую функцию, мы можем за O(1) обработать элемент.

Естественно, возникает вопрос, почему невозможно такое, что мы попадем дважды в одну ячейку массива, ведь представить функцию, которая ставит в сравнение каждому элементу совершенно различные натуральные числа просто невозможно. Именно так возникает проблема коллизии или проблема, когда хеш-функция выдает одинаковое натуральное число для разных элементов.

Существует несколько решений данной проблемы: метод цепочек и метод двойного хеширования, который и был использован в данной работе.

При двойном хешировании используются две независимые хеш-функции . Пусть — это наш ключ, — размер нашей таблицы, — остаток от деления на , тогда сначала исследуется ячейка с адресом , если она уже занята, то рассматривается , затем и так далее. В общем случае идёт проверка последовательности ячеек где

Таким образом, операции вставки, удаления и поиска в лучшем случае выполняются за O(1), в худшем — за O(m), что не отличается от обычного линейного разрешения коллизий. Однако в среднем, при грамотном выборе хеш-функций, двойное хеширование будет выдавать лучшие результаты, за счёт того, что вероятность совпадения значений сразу двух независимых хеш-функций ниже, чем одной.

# Листинг программы

Программа расположена на крупнейшем веб-сервисе для хостинга IT-проектов и их совместной разработки «GitHub».

Ссылка на репозиторий:

<https://github.com/qkonstantin/DoubleHashtableGUI>

# Работа программы

При запуске программы отображается окно с пустой таблицей (рис. 1), в которую мы можем:

* добавить пары «ключ-значение»
* получить значение выбранного ключа
* проверить содержание ключа в таблице
* удалить выбранную пару

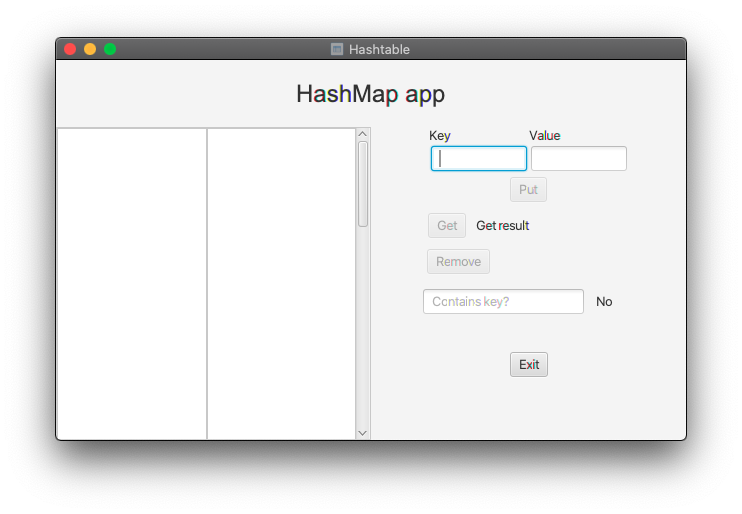


Рис. 1 Начальное окно программы

Введем в поля значения «key» и «value» и нажмем кнопку «Put» (рис. 2–3).

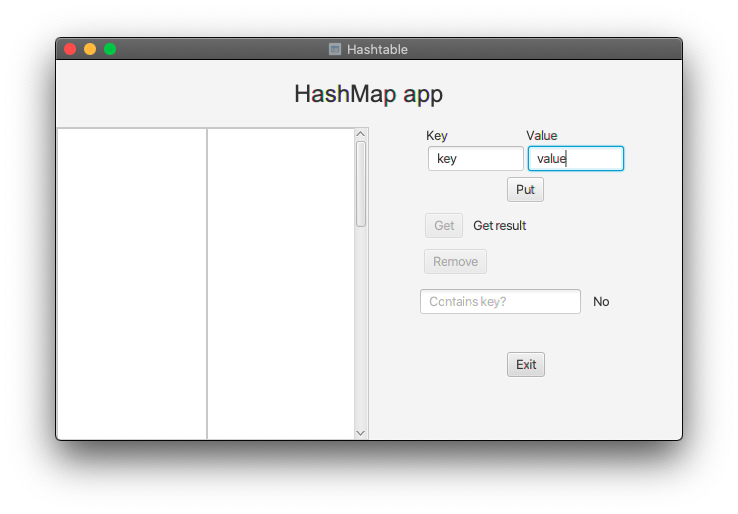


Рис. 2 Ввод данных

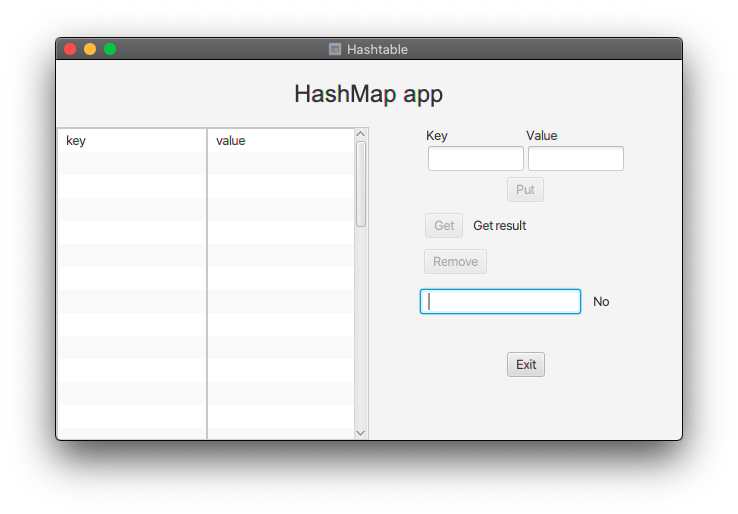


Рис. 3 Отображение данных в таблице

Кликнем по ключу «key» и нажмем «Get». Получаем напротив значение – «value» (рис. 4).

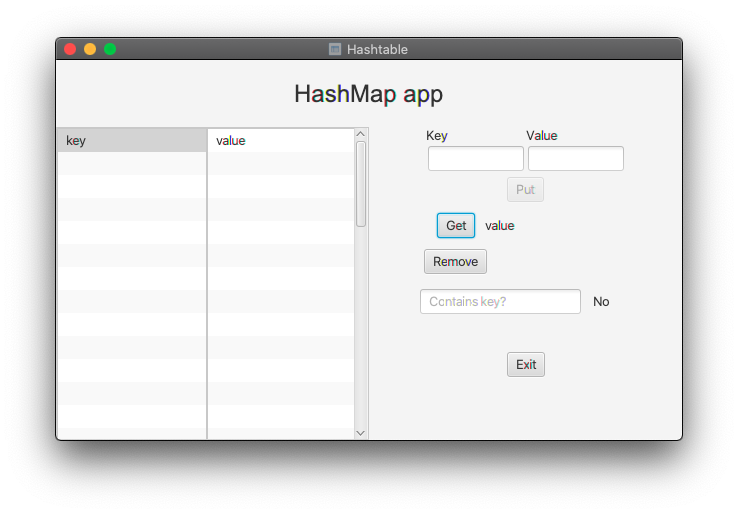


Рис. 4 Получение значения по выбранному ключу

В поле «Contains key?» введем ключ «key». Получаем ответ, что такой ключ содержится в таблице (рис. 5).

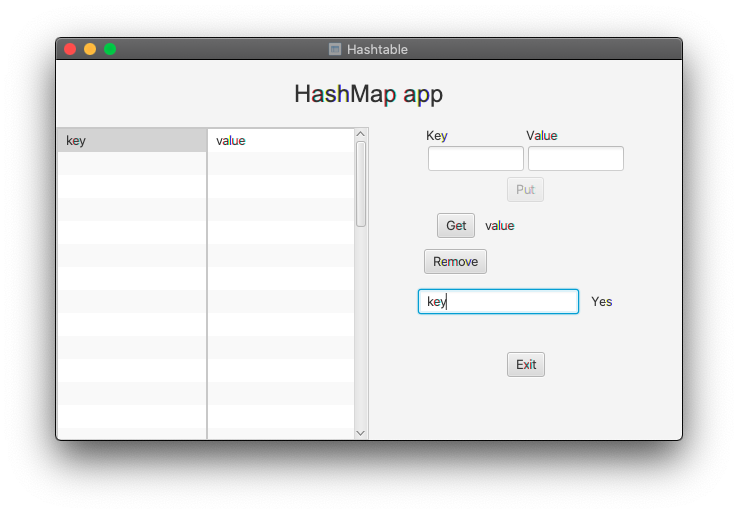


Рис. 5 Проверка на вхождения ключа в таблицу

Нажмем на кнопку «Remove», тем самым удалив выбранную пару. В поле «Contains key?» получаем ответ, что такого ключа в таблице нет (рис. 6).

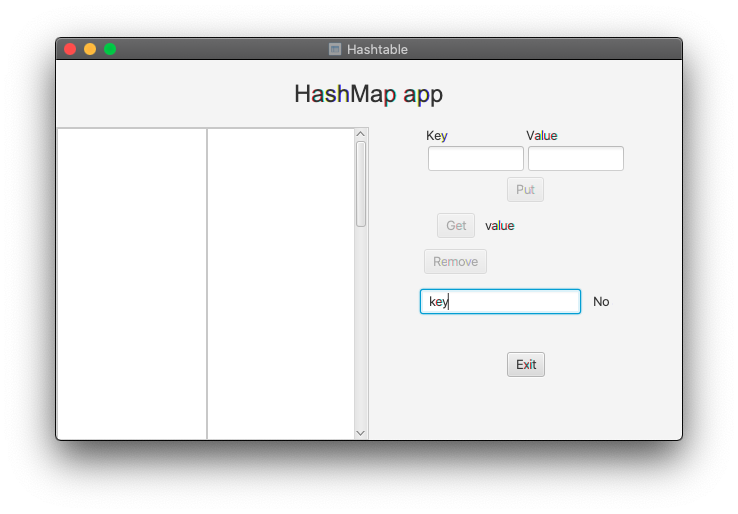


Рис. 6 Проверка на вхождения ключа в таблицу

Хеш-таблица может хранить данные типа «String». Ниже продемонстрировано несколько вариантов нормальной работы (рис 7-10).

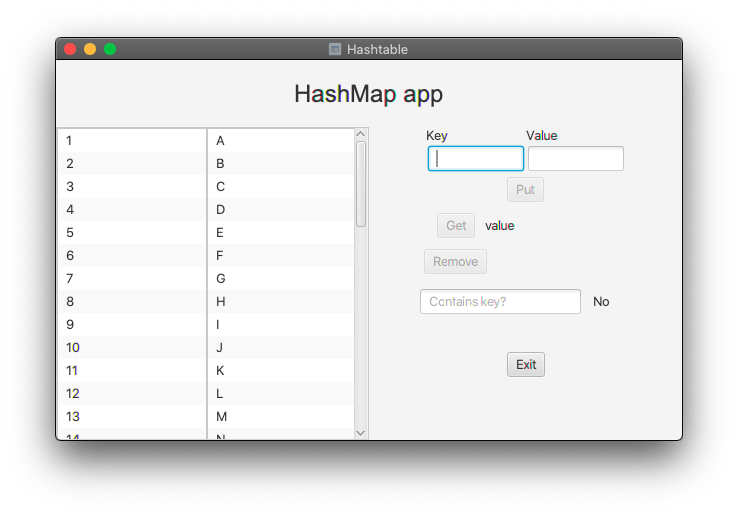


Рис. 7 Пример работы программы

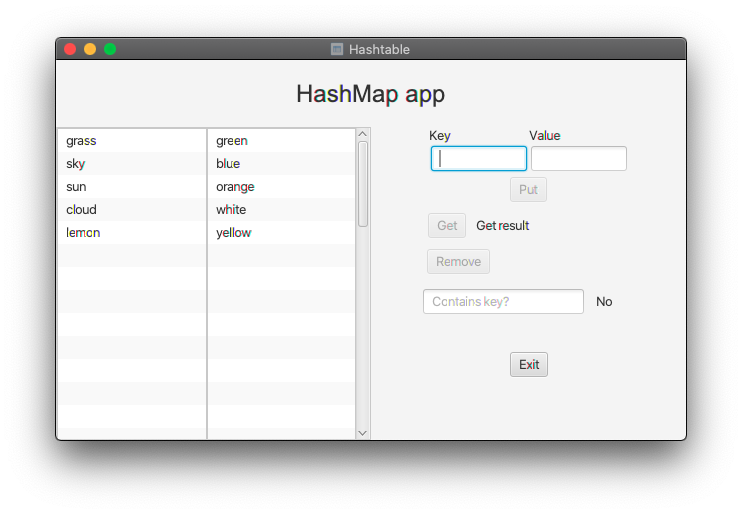


Рис. 8 Пример работы программы

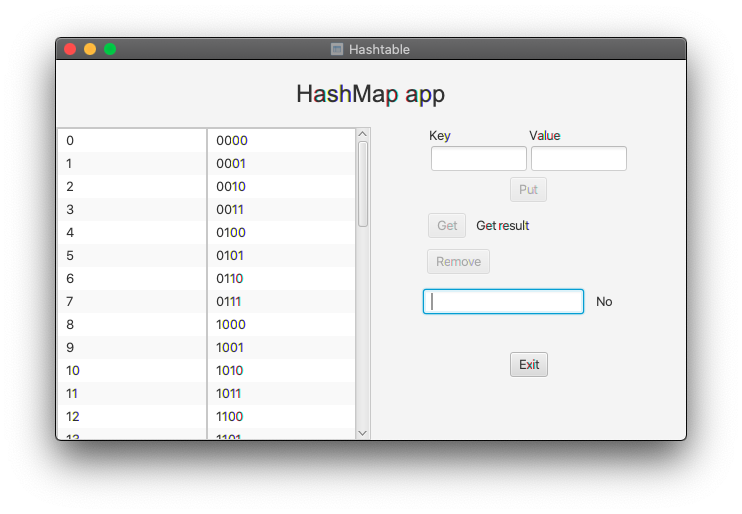


Рис. 9 Пример работы программы

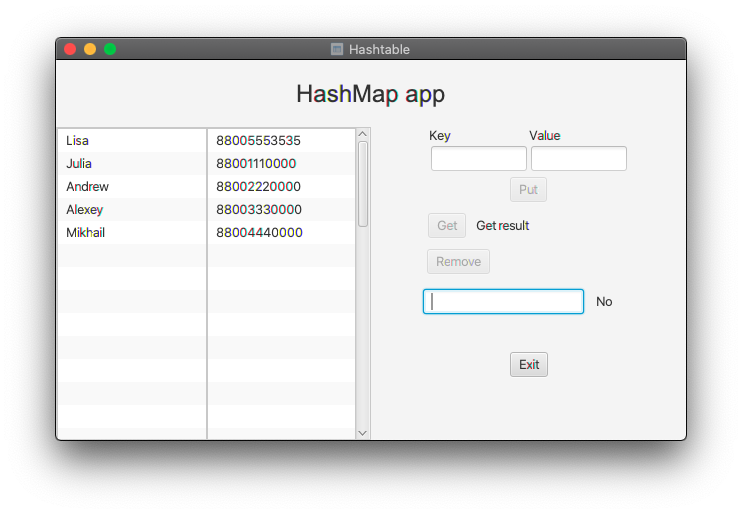


Рис. 10 Пример работы программы