Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**КУРСОВОЙ ПРОЕКТ**

**Хеш-таблица с открытой адресацией**

по дисциплине «Алгоритмы и структуры данных»

Выполнил студент гр.3530901/80002 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Прозоров Ф. А.

(подпись)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гагарский К. А.

(подпись)

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 года

Санкт-Петербург   
2019Санкт-Петербургский политехнический университет Петра Великого

**ЗАДАНИЕ**

**НА ВЫПолнение курсовоГО ПРОЕКТА**

студенту группы 3530901/80002 Прозорову Филиппу Антоновичу

(номер группы) (ФИО)

***1*. Тема проекта:** Хеш-таблица с открытой адресацией.

**2. Срок сдачи законченной работы:**

***3*. Исходные данные к работе**: Intellij IDEA Community Edition 2019.2.2, Java JDK 1.8

***4.*Содержание пояснительной записки**: введение, основная часть (текст программы, описание программы, испытания программы, заключение, список использованных источников).

**Дата получения задания:** «     »         2019 г.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гагарский К. А.

(подпись)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Прозоров Ф.А

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

# Содержание

[Содержание 3](#_Toc28293969)

[Введение 4](#_Toc28293970)

[Текст программы 4](#_Toc28293971)

[Описание проекта 5](#_Toc28293972)

[Проверка работоспособности 5](#_Toc28293973)

[Заключение 7](#_Toc28293974)

[Список использованных источников 7](#_Toc28293975)

# Введение

Исходный текст задания: «Реализация в виде класса на Java одного из видов хеш-таблицы, например приведённые ниже (темы имеют среднюю сложность). 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). Для усложнения задачи в неё можно добавить визуализацию». Выбранный для выполнения вариант хеш-таблицы: хеш-таблица с открытой адресацией.

# Текст программы

Текст программы содержится в репозитории, расположенном на веб-сервисе Github:

<https://github.com/PhilippProzorov/Hash-Table>

# Описание проекта

В массиве *H* хранятся сами пары ключ-значение. Алгоритм вставки элемента проверяет ячейки массива *H* в некотором порядке до тех пор, пока не будет найдена первая свободная ячейка, в которую и будет записан новый элемент. Этот порядок вычисляется на лету, что позволяет сэкономить на памяти для указателей, требующихся в хеш-таблицах с цепочками.

Последовательность, в которой просматриваются ячейки хеш-таблицы, называется **последовательностью проб**. В общем случае, она зависит только от ключа элемента, то есть это последовательность *h*0(*x*), *h*1(*x*), …, *hn*— 1(*x*), где *x* — ключ элемента, а *hi*(*x*) — произвольные функции, сопоставляющие каждому ключу ячейку в хеш-таблице. Первый элемент в последовательности равен значению некоторой хеш-функции от ключа, а остальные считаются от него одним из приведённых ниже способов. Для успешной работы алгоритмов поиска последовательность проб должна быть такой, чтобы все ячейки хеш-таблицы оказались просмотренными ровно по одному разу.

Алгоритм поиска просматривает ячейки хеш-таблицы в том же самом порядке, что и при вставке, до тех пор, пока не найдется либо элемент с искомым ключом, либо свободная ячейка (что означает отсутствие элемента в хеш-таблице).

Проект состоит из следующих элементов:

1. **Package main**

В данном пакете содержится реализация хеш-таблицы с открытой адресацией.

**HashTable.java**

Основное тело проекта, где приведены методы работы с хеш-таблицей и обработка ситуаций, которые могут произойти во время работы с ней.

**HashEntry.java**

Небольшой класс, занимающийся взаимодействием с Key и Value элемента (Пары «Ключ и значение» соответственно). Реализованные методы: getKey, getValue, setValue.

1. **Package test**

В этом пакете содержатся все применимые для хеш-таблицы тесты.

# Проверка работоспособности

Для проверки работоспособности проекта были написаны тесты, проверяющие, отвечает ли хеш-таблица требованиям задания. Ниже представлен полный список тестов из класса Tests:

1. checkInput()
2. remove()
3. addHighQuantity()
4. removeHighQuantity()
5. checkCollisions()
6. returnNull()
7. returnOld()

Написанный класс проходит все 7 тестов, возвращая корректные результаты.

# Заключение

В результате разработки получилась работающая версия хеш-таблицы с открытой адресацией. В ходе работы были получены знания о порядках переборов элементов в таблицах, формах реализации структур данных, и алгоритмов, работающих с ними. В дальнейшем, проект можно улучшить: добавить визуализацию, возможно, усовершенствовать алгоритмы обработки, увеличить количество тестов, через которые может пройти проект.

# Список использованных источников

1. Open addressing

<https://en.wikipedia.org/wiki/Open_addressing>

1. Hash Table. Open addressing strategy

<http://www.algolist.net/Data_structures/Hash_table/Open_addressing>

1. What is open addressing?

<https://jameshfisher.com/2017/12/10/what-is-open-addressing/>

1. [Open Addressing | Linear Probing | Collision](https://www.gatevidyalay.com/open-addressing-collision-resolution-techniques/)

<https://www.gatevidyalay.com/open-addressing-collision-resolution-techniques/>

### Open Addressing: Handling collision in hashing

[http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/Map/open- addr.html](http://www.mathcs.emory.edu/~cheung/Courses/323/Syllabus/Map/open-%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20%20addr.html)

Также информация об открытой адресации была взята с сайта:

[https://stackoverflow.com/](https://stackoverflow.com/questions/9127207/hash-table-why-deletion-is-difficult-in-open-addressing-scheme)