Минобрнауки россии

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”

(ФГБОУ ВО «ВГУ»)

Факультет компьютерных наук

Реализация структуры данных HashMap на языке Java и ее отображение в виде графа в формате Graphviz

Проектная работа

4 семестр 2024/2025 учебного года

09.03.02 Информационные системы и технологии

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. 2 курса Е.А. Аверкиева

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преп. Н.К. Самойлов

Воронеж 2025

**ВВЕДЕНИЕ**

Современные вычислительные задачи требуют эффективных структур данных, обеспечивающих быстрый доступ, вставку и удаление элементов. Одной из таких структур является HashMap. Она позволяет хранить пары "ключ-значение" и обеспечивает быстрый доступ к данным за счет хеширования. В языке Java HashMap реализован как часть стандартной библиотеки (в пакете java.util) и широко применяется в задачах, где требуется высокая производительность операций вставки, удаления и поиска элементов.

Однако для глубокого понимания работы HashMap важно не только знать его интерфейс, но и представлять внутреннюю организацию, включая принцип хеширования, разрешение коллизий и динамическое расширение таблицы. Визуализация структуры данных в виде графа позволяет наглядно продемонстрировать эти механизмы, упрощая анализ и отладку.

Инструмент Graphviz предоставляет удобный способ генерации таких графических представлений, позволяя наглядно анализировать распределение ключей, разрешение коллизий (например, через цепочки узлов) и изменение структуры при рехешировании.

1. **Постановка задачи**

Основной целью проектной работы является реализация структуры данных HashMap на языке программирования Java и ее отображение в виде графа в формате Graphviz. Основные требования к создаваемой программе формулируются следующим образом:

* Изучение принципов работы HashMap.
* Реализация собственной версии HashMap на Java.
* Исследование работы HashMap при различных сценариях.
* Визуализация структуры данных HashMap в виде графа с помощью Graphviz.

1. **Теоретические основы HashMap**

**Хеш-таблицей (**HashMap) называется структура данных, реализующая интерфейс **ассоциативного массива** (абстрактная модель «ключ – значение» или entry), которая обеспечивает очень быструю вставку и поиск: независимо от количества элементов вставка и поиск (а иногда и удаление) выполняются за время, близкое к константе – O(1). По сути, это обычный массив, где местоположение элемента зависит от значения самого элемента. **Связь между значением элемента и его позицией в хеш-таблице задает хеш-функция.**

**Хеш-функция** получает входную часть данных, которую мы называем **ключом**, а на выходе она выдает целое число, известное как **хеш-значение (или хеш-код)**. Затем, **хеш-значение** привязывает наш ключ к определенному индексу хеш-таблицы. Для основных операций: вставки, поиска и удаления мы используем одну и ту же хеш-функцию, поэтому эти операции осуществляются довольно быстро.

1. **Реализация**
   1. **Средства реализации**

В качестве средств реализации для разработки программы для наглядной демонстрации структуры данных HashMap использованы:

* объектно-ориентированный язык Java,
* пакет java.util,
* модуль JUnit Jupiter,
* пакет Graphviz.
  1. **Логика приложения**

Разработанная программа состоит из следующих классов:

* Main
* HashMap;
* HashMapGraph;
* HashMapTest.  **???**

**3.2.1 Класс Main**

Класс Main является основным классом данной программы. В нем находится точка входа в программу. Данный класс демонстрирует работу класса HashMap и вызывает визуализацию HashMap через Graphviz.

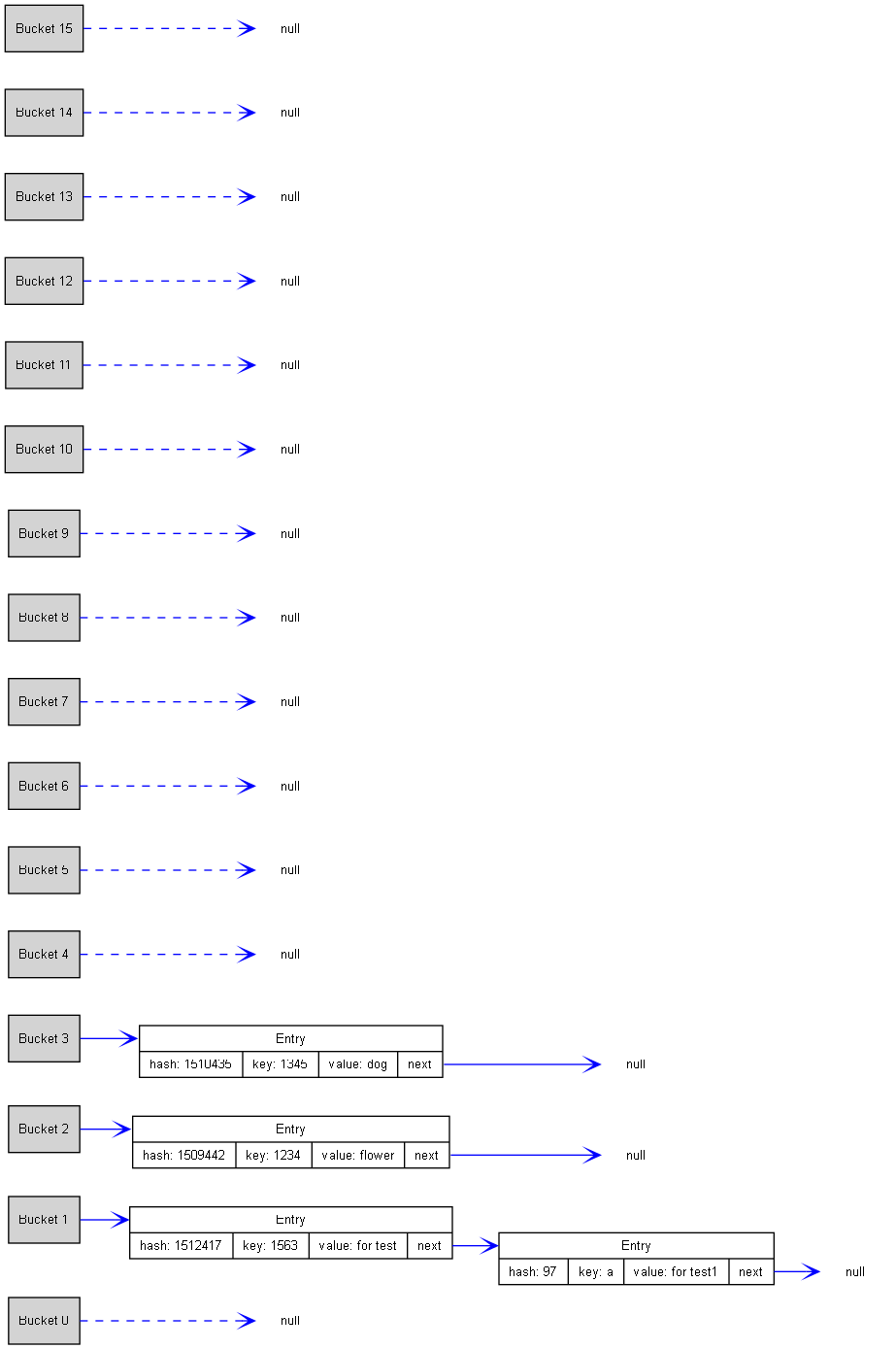
**3.2.2 Класс HashMap**

Данный класс предоставляет стандартный интерфейс структуры данных HashMap. включая операции добавления, получения и удаления элементов. Внутренняя структура данных представляет собой массив связных списков, где каждый список соответствует определённой корзине (бакету) в хеш-таблице. При добавлении элементов вычисляется хеш-код ключа, который определяет индекс корзины. Если в одной корзине оказывается несколько элементов (коллизия), они хранятся в виде цепочки узлов.

Класс содержит основные методы работы с хеш-таблицей, такие как put, get, removeKey, containsKey, hash, и тд. Данные методы обеспечивают выполнение различных операций со структурой данных, таких как добавление, получение и удаление элементов.

**3.2.3 Класс HashMapGraph**

Класс HashMapGraph предназначен для визуализации структуры хеш-таблицы. Он генерирует описание графа в формате DOT, который затем может быть преобразован в графическое изображение с помощью утилиты Graphviz. Визуализация отображает все корзины таблицы, цепочки коллизий, а также ключи и значения элементов. Это позволяет наглядно анализировать распределение элементов.



**3.2.4 Класс HashMapTest**

Тестирование функциональности осуществляется с помощью класса HashMapTest, содержащего набор юнит-тестов. Тесты проверяют основные сценарии работы: добавление и обновление элементов, удаление, обработку коллизий, корректность подсчёта размера таблицы.

**ЗАКЛЮЧЕНИЕ**

В результате выполненной проектной работы были изучены принципы работы HashMap. Была реализована собственная версия HashMap на языке программирования Java, а также сделана ее визуализация в виде графа с помощью Graphviz.