МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего профессионального образования

**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра прикладной математики, механики, управления и программного обеспечения**

**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ**

по дисциплине «Структуры и алгоритмы компьютерной обработки данных»

Направление «Математическое обеспечение и администрирования информационных систем»

|  |
| --- |
| Выполнил студент гр. Б8204  \_\_\_\_\_\_\_\_\_\_Слипенчук Дмитрий |
| Проверил доцент, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_ С.Н. Остроухова  (зачтено/не зачтено) |

г. Владивосток

2015

**1. Неформальная постановка задачи**

Разработать класс хеш-таблиц для работы со списком записей, обладающих следующими полями:

* Emei телефона
* название телефона

Класс должен содержать в себе следующие методы:

1. Первичная функция хеширования;

2. Вторичная функция хеширования;

3. Добавление записи;

4. Удаление записи;

5. Поиск записи;

6. Печать всей таблицы;

7. Конструктор;

8. Деструктор;

2. Спецификация методов класса

**hash\_table (int size);**

Конструктор создает таблицу, обладающую заданным размером, иначе при отсутствии явного задания размера создать таблицу, состоящую из 9 записей. При задании недопустимой размерности таблицы создается таблица, состоящая из 9 записей.

**~ hash\_table ();**

Деструктор должен освобождать память, выделенную под таблицу, созданную конструктором.

**void write();**

Данный метод должен выводить в терминал все записи таблицы.

**int hash\_func\_1(int key);**

Данный метод вычисляет позицию **key** с помощью первичной функции хеширования и возвращает номер этой позиции. Позиция вычисляет по формуле: key mod размерность таблицы.

**int hash\_func\_2(int key);**

Данный метод вычисляет позицию **key** с помощью вторичной функции хеширования и возвращает номер этой позиции. Позиция вычисляет по формуле: ( key + 1 ) mod размерность таблицы.

**int add(line DB);**

Данный метод добавляет запись DB в таблицу. Возвращает **true**, если добавление прошло успешно, и **false**, если таблица полностью заполнена или запись DB уже добавлена или запись с таким DB.emei уже находится в тадлице.

**bool search(line DB);**

Данный метод ищет запись DB в таблице и возвращает **true**, если поиск удачен, и **false**, если запись не найдена.

**bool del(line DB);**

Данный метод удалает запись DB из таблицы. Возвращает **false**, если элемента нет в таблице, и **true** при удачном удалении.

**3. Тестирование**

**Тестирование метода hash\_table (int size);**

|  |  |  |
| --- | --- | --- |
| **Описание теста** | **Входные данные** | **Результат** |
| Создание таблицы с явным заданием размера(5) |  | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 |
| Создание таблицы с заранее заданым размером |  | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 |
| Создание таблицы недопустимого размера (-5) |  | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 |

**Тестирование метода int add(line DB);**

|  |  |  |  |
| --- | --- | --- | --- |
| Описание теста | Исходная таблица | Входные данные | Результат |
| Добавление элемента в пустую хеш таблицу | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 | emei = 12;  name = “galaxy”; | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = 12 name = galaxy status = 1  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 |
| Добавление элемента в заполненную хеш таблицу | emei = 25 name = sony status = 1  emei = 25 name = iphone status = 1  emei = 25 name = philips status = 1  emei = 12 name = galaxy status = 1  emei = 21 name = htc status = 1  emei = 30 name = wileyfox status = 1  emei = 24 name = meizu status = 1  emei = 25 name = xiaomi status = 1  emei = 25 name = lg status = 1 | emei = 24;  name = "dexp"; | emei = 25 name = sony status = 1  emei = 25 name = iphone status = 1  emei = 25 name = philips status = 1  emei = 12 name = galaxy status = 1  emei = 21 name = htc status = 1  emei = 30 name = wileyfox status = 1  emei = 24 name = meizu status = 1  emei = 25 name = xiaomi status = 1  emei = 25 name = lg status = 1  ERROR: dexp NOT ADD |
| Добавление элемента в не полностью заполненной хеш таблице |  | emei = 25;  name = "philips"; | emei = 25 name = sony status = 1  emei = 25 name = iphone status = 1  emei = 25 name = philips status = 1  emei = 12 name = galaxy status = 1  emei = 21 name = htc status = 1  emei = 30 name = wileyfox status = 1  emei = 24 name = meizu status = 1  emei = 25 name = xiaomi status = 1  emei = 25 name = lg status = 1 |
| Добавление элемента с возникновением коллизии |  | emei = 12;  name = "galaxy";  emei = 21;  name = "htc"; | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = 12 name = galaxy status = 1  emei = 21 name = htc status = 1  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0 |
| Добавление элемента в конец списка с возникновением коллизии |  | emei = 25;  name = "xiaomi";  emei = 25;  name = "lg";  emei = 25;  name = "sony"; | emei = 25 name = sony status = 1  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = 25 name = xiaomi status = 1  emei = 25 name = lg status = 1 |
| Добавление элемента на место удалённого элемента с помощью вторичной функции хеширования |  | emei = 25;  name = "xiaomi";  emei = 25;  name = "lg";  emei = 25;  name = "sony"; | emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = 25 name = lg status = 1  emei = 25 name = sony status = 1 |
| Добавление в таблицу повторяющегося элемента |  | Terminator 1992 1  Terminator 2 1996 1  Garfild 2005 1  0 0 | emei = 25 name = sony status = 1  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = name = status = 0  emei = 25 name = xiaomi status = 1  emei = 25 name = lg status = 1  ERROR: xiaomi have already been added. |

**Тестирование метода bool search(picture film);**

|  |  |  |
| --- | --- | --- |
| Описание теста | Входные данные | Результат |
| Поиск элемента в пустой хеш таблице | emei = 24;  name = "dexp"; | false |
| Поиск элемента в заполненной хеш таблице | emei = 24;  name = "dexp"; | true |
| Поиск элемента в не полностью заполненной хеш таблице | emei = 24;  name = "dexp"; | true |

**Тестирование метода bool del(picture film);**

|  |  |  |
| --- | --- | --- |
| Описание теста | Входные данные | Результат |
| Удаление элемента в пустой хеш таблице | emei = 25;  name = "xiaomi"; | ERROR: xiaomi NOT DELL |
| Удаление элемента в заполненной хеш таблице | emei = 25;  name = "xiaomi"; | true |
| Удаление ранее удалённого элемента | emei = 25;  name = "xiaomi"; | ERROR: xiaomi NOT DELL |