МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

профессионального образования

**«Дальневосточный федеральный университет»**

**ШКОЛА ЕСТЕСТВЕННЫХ НАУК**

**Кафедра информационных систем управления**

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 4**

По дисциплине «Методы, средства и технологии информационных систем управления»

На тему «Хеширование, хеш-таблицы»

|  |
| --- |
| Выполнил студент гр. Б8219 |
| О.В. Константинов |
|  |
| Проверил ст. преп. |
| Е.А. Елсукова |
|  |
| (зачтено/не зачтено) |

г. Владивосток

2017

# Аннотация

Данный отчет подготовлен в рамках задания по предмету «Методы, средства и технологии информационных систем управления». Отчет призван систематизировать знания о методах работы с хеш-таблицами. Структура документа разделена, в соответствии с требованием преподавателя, на несколько разделов: «Постановка задачи», «Функциональное описание», «Примеры выполнения программы», «Вывод по работе».

# Постановка задачи

Требуется разработать демонстрационную программу, реализующую следующие методы работы с хеш-таблицами:

* Добавление элемента.
* Удаление элемента.
* Поиск элемента по ключу.

В качестве входных данных требуется использовать поля структуры PROFIT.

Результат работы программы должен представлять собой визуализацию хеш-таблицы.

# Функциональное описание

Программа представляет собой оконное приложение для ОС Windows NT (не ниже 5.1). Приложение было реализовано в IDE Visual Studio версии 2015, на языке C#.

Структурно, проект состоит из одной формы: MainWindow.xaml.cs. И пяти модулей: ListViewData.cs, HashTableNodePair.cs, HashTableArrayNode.cs, HashTableArray.cs, HashTable.cs.

В программе, данные представляют собой последовательность объектов класса ListViewData:

public class ListViewData // Идентичен с PROFIT

{

public ListViewData(string id, string name, string count)

{

Id = id;

Name = name;

Count = count;

}

public string Id { get; set; } // Код товара

public string Name { get; set; } // Наименование товара

public string Count { get; set; } // Количество продаж

}

Значения хранятся в ячейках массива двусвязным списком при помощи класса HashTableNodePair<TKey, TValue>, где TKey – ключ (String), TValue – значение (ListViewData):

public class HashTableNodePair<TKey, TValue>

{

public HashTableNodePair(TKey key, TValue value)

{

Key = key;

Value = value;

}

public TKey Key

{

get;

private set;

}

public TValue Value

{

get;

set;

}

}

Сами же ячейки массива представляют собой класс HashTableArrayNode<TKey, TValue>:

public class HashTableArrayNode<TKey, TValue>

{

LinkedList<HashTableNodePair<TKey, TValue>> \_items;

public void Add(TKey key, TValue value);

public void Update(TKey key, TValue value);

public bool TryGetValue(TKey key, out TValue value);

public bool Remove(TKey key);

public void Clear();

public IEnumerable<HashTableNodePair<TKey, TValue>> Items;

}

Массив из двусвязных списков реализуется посредством класса HashTableArray<TKey, TValue>, над ним определены основные методы работы с массивами:

public class HashTableArray<TKey, TValue> where TValue : class

{

HashTableArrayNode<TKey, TValue>[] \_array;

public HashTableArray(int capacity);

private int GetIndex(TKey key);

public int Capacity;

public void Add(TKey key, TValue value);

public bool Remove(TKey key);

public TValue Contains(TKey key);

public IEnumerable<HashTableNodePair<TKey, TValue>> Items;

}

Управление размером происходит через класс HashTable<TKey, TValue>:

public class HashTable<TKey, TValue> where TValue : class

{

// Если заполнение массива превышает 75% - массив увеличивается.

const double \_fillFactor = 0.75;

int \_maxItemsAtCurrentSize; // Максимальая вместимость массива

int \_count; // Количество элементов

HashTableArray<TKey, TValue> \_array;

public HashTable() : this(100);

public HashTable(int initialCapacity);

public void Add(TKey key, TValue value);

public bool Remove(TKey key);

public TValue ContainsKey(TKey key);

public IEnumerator<HashTableNodePair<TKey, TValue>> GetEnumerator();

public int Count;

}

В нынешней реализации данные хранятся в файле ListViewData.xml. Запрос на сохранение значений производится при закрытии программы.

# Примеры выполнения программы

ВХОД (ПОСЛЕДОВАТЕЛЬНОСТЬ СТРУКТУР):

49788176, SCANIA 114 {05-} ФАРА ЛЕВ, 223

45789959, SCANIA 114 ФАРА ПРОТИВОТУМ ПРАВ, 384

82188843, SCANIA 114 ФОНАРЬ ЗАДН ВНЕШН ЛЕВ, 232

71045012, FE/FL ФАРА ПРАВ П/КОРРЕКТОР, 711

50226014, FH/FM ФАРА ПРАВ П/КОРРЕКТОР С УНИВЕР., 891

41463790, FH/FM ФАРА ПРОТИВОТУМ П, 131

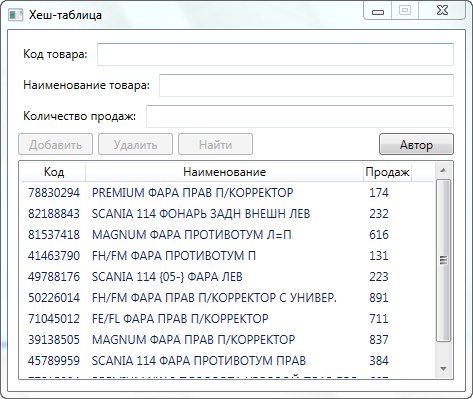
77215904, PREMIUM УКАЗ.ПОВОРОТА УГЛОВОЙ ПРАВ БЕЛ, 607

78830294, PREMIUM ФАРА ПРАВ П/КОРРЕКТОР, 174

39138505, MAGNUM ФАРА ПРАВ П/КОРРЕКТОР, 837

81537418, MAGNUM ФАРА ПРОТИВОТУМ Л=П, 616

ВЫХОД (ВИЗУАЛИЗИРОВАННАЯ ХЕШ-ТАБЛИЦА):



ВХОД (ЗАПРОС НА УДАЛЕНИЕ СТРУКТУР ПО КОДУ):

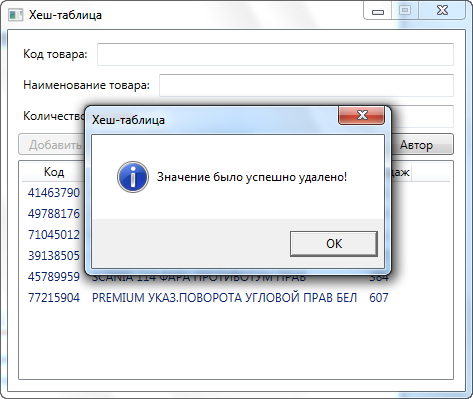
78830294

82188843

81537418

50226014

ВЫХОД (ВИЗУАЛИЗИРОВАННАЯ ХЕШ-ТАБЛИЦА):



# Вывод по работе

В результате проведенной работы была достигнута поставленная цель: изучение основных методов работы с хеш-таблицами.