Лабораторная работа №3

Вариант 16

Салихов А.Г. гр. Б04-191-1з

1.Постановка задачи.

Хэш-таблица должна поддерживать следующие операции:

1) Инициализация

2) Вставка элемента (целочисленный ключ + значение)

3) Получение значения элемента по ключу

4) Удаление элемента по ключу

5) Освобождение памяти от структуры данных

2.Исходный код решения

#include <cstdlib>

#include "stdio.h"

#include <string>

#include <iostream>

using namespace std;

struct Data {

int value;

string key = "";

};

struct HashTable {

int size = 0;

int elements = 0;

Data\* array= NULL;

void init() {

if (array) return;

array = new Data[3];

size = 3;

}

int h(string key) {

int hash = 0;

for (int i = 0; i < key.size(); i++) {

hash = hash + (int)key[i];

}

hash = hash % size;

return hash;

}

void add(int value, string key) {

if (get(key)) return;

Data\* data = new Data();

data->value = value;

data->key = key;

if (elements + 1 > (size / 2)) {

resize();

}

int hash = h(key);

while (array[hash].key != "") {

if (array[hash].key == key) return;

hash++;

if (hash == size) hash = 0;

}

array[hash] = \*data;

elements++;

}

bool get(string key, int\* val) {

Data\* getData = get(key);

if (getData) {

\*val = getData->value;

return true;

}

else return false;

}

Data\* get(string key) {

if (key == "") return NULL;

int hash = h(key);

int count = 0;

while (count < size) {

if (array[hash].key == key) {

return &array[hash];

}

hash++;

hash = hash % size;

count++;

}

return NULL;

}

void resize() {

Data\* oldArray = array;

array = new Data[size \* 2];

int oldSize = size;

size = size \* 2;

elements = 0;

for (int i = 0; i < oldSize; i++) {

if (oldArray[i].key != "") {

add(oldArray[i].value, oldArray[i].key);

}

}

delete[] oldArray;

}

void remove(string key) {

Data\* getData = get(key);

if (getData) {

getData->key = "";

elements--;

}

}

void removeFromMemory() {

delete[] array;

size = 0;

elements = 0;

array = NULL;

}

void printAll() {

if (size == 0) {

printf("Table is EMPTY.\n");

return;

}

for (int i = 0; i < size; i++) {

if (array[i].key != "") {

cout << array[i].key << " = " << array[i].value <<endl;

}

}

cout << endl;

}

};

int main() {

HashTable q;

q.init();

q.add(1, "key1");

q.add(2, "key2");

q.add(3, "key3");

q.printAll();

Data\* q1 = q.get("key2");

cout << "Poluchennoe znachenie = " << q1->value << endl;

q.remove("key2");

q.printAll();

q.removeFromMemory();

q.printAll();

}

3.Контрольный пример

