

ГУАП

КАФЕДРА № 43

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

ассистент

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

К.А. Кочин

\_\_\_\_\_  
инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

ХЕШИРОВАНИЕ ДАННЫХ

по курсу: Структуры и алгоритмы обработки данных

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

4136

\_\_\_\_\_  
подпись, дата

Бобрович Н. С.

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2022

## Цель работы

Целью работы является изучение методов хеширования данных и получение практических навыков реализации хеш-таблиц.

## Задание

Вариант №18.

Составить хеш-функцию в соответствии с заданным вариантом и проанализировать ее. При необходимости доработать хеш-функцию. Используя полученную хеш-функцию разработать на языке программирования высокого уровня программу, которая должна выполнять следующие функции:

1. создавать хеш-таблицу;
2. добавлять элементы в хеш-таблицу;
3. просматривать хеш-таблицу;
4. искать элементы в хеш-таблице по номеру сегмента/по ключу;
5. выгружать содержимое хеш-таблицы в файл для построения гистограммы в MS Excel, или в аналогичном подходящем ПО;
6. удалять элементы из хеш-таблицы; в программе должна быть реализована проверка формата вводимого ключа;

7. при удалении элементов из хэш-таблицы, в программе должен быть реализован алгоритм, позволяющий искать элементы, вызвавшие коллизию с удаленным;
8. в программе должен быть реализован алгоритм, обрабатывающий ситуации с переполнением хэш-таблицы.

ЦцББцц; 2000; линейное опробование.

## **Листинг программы**

```

1  #include <iostream>
2  #include<string>
3  #include <cctype>
4  #include <fstream>
5  #include <ctime>
6  #include <regex>
7
8  using namespace std;
9
10 const int c = 1, N = 2000;
11 const int key_lenght = 6;
12
13 struct Elem {
14     string key;
15     bool deleted = false;
16 };
17
18
19 class Table {
20     Elem dict[N];
21     bool is_key(string key) {
22         static const regex r(R"^[0-9][0-9][A-Z][A-Z][0-9][0-9]$");
23         return regex_match(key.data(), r);
24     }
25     int hash(string key) {
26         return (char(key[0]) + char(key[1]) + char(key[2]) + char(key[3]) + char(key[4]) + char(key[5]));
27     }
28     int ahash(string key) {
29         return int(key[0]) + int(key[1]) + int(key[2]) + int(key[3]) + int(key[4]) + int(key[5]);
30     }
31
32 public:
33     void add(string key) {
34         if (!is_key(key)) {
35             cout << "Ошибка ";
36             return;
37         }
38         int id;
39         for (int i = 0;; i++) {
40             id = hash(key) + c * i;
41             if (id >= N) {
42                 return;
43             }
44             if (dict[id].key.empty()) {
45                 dict[id].key = key;
46                 return;
47             }
48         }

```

```

49     }
50
51     int find(string key) {
52         if (!is_key(key)) {
53             cout << "\nError!\n";
54             return -1;
55         }
56         int id;
57         for (int i = 0;; i++) {
58             id = hash(key) + c * i;
59             if (id >= N || (!dict[id].deleted && dict[id].key.empty())) {
60                 return -1;
61             }
62             if (dict[id].key == key) {
63                 return id;
64             }
65         }
66     }
67
68     int afind(string key) {
69         int id = ahash(key);
70         return id;
71     }
72

```

```

73     string find(int index) {
74         if (index >= N) {
75             cout << "Error!";
76             return "";
77         }
78         if (dict[index].key.empty())
79             return "";
80
81         return dict[index].key;
82     }
83
84     void show() {
85         for (int i = 0; i < N; i++)
86             if (!dict[i].key.empty())
87                 cout << i << " " << dict[i].key << " ";
88         cout << endl;
89     }
90
91     void del(string key) {
92         if (!is_key(key)) {
93             cout << "\nError!\n";
94             return;
95         }
96         int id = find(key);

```

```

97         if (id != -1) {
98             dict[id].deleted = true;
99             dict[id].key = "";
100         }
101     }
102 };
103
104 void fill_table(Table& table) {
105     string key = "123456";
106     for (int i = 0; i < N * 2; i++) {
107         for (int j = 0; j < key_lenght; j++) {
108             if (j == 2 || j == 3)
109                 key[j] = 'A' + rand() % ('Z' - 'A');
110             else
111                 key[j] = '0' + rand() % ('9' - '0');
112         }
113         table.add(key);
114     }
115 }
116
117 void w_file(Table table) {
118     fstream file("smth.txt", ios_base::out);
119     if (!file.is_open()) {
120         cout << "Error!\n";

```

```

121         return;
122     }
123     for (int i = 0; i < N; i++) {
124         if (!table.find(i).empty())
125             file << i << "\t" << table.afind(table.find(i)) << '\n';
126     }
127     file.close();
128 }
129
130 string input_str() {
131     cout << "Введите название строки: ";
132     string value;
133     cin >> value;
134     return value;
135 }
136
137 int input_int() {
138     int value;
139     cin >> value;
140     return value;
141 }
142
143 int main()
144 {

```



```

145     setlocale(0, "Rus");
146     srand(time(NULL));
147     int c = 9;
148     Table table;
149     while (c != 0) {
150         cout << endl;
151         cout << "1, если нужно заполнить таблицу.\n";
152         cout << "2, если нужно вывести таблицу.\n";
153         cout << "3, если нужно добавить элемент.\n";
154         cout << "4, если нужно удалить элемент по названию.\n";
155         cout << "5, если нужно найти элемент по названию.\n";
156         cout << "6, если нужно перенести таблицу в файл.\n";
157         cout << "0, если ничего больше не нужно.\n";
158         cout << endl;
159         c = input_int();
160         if ((c != 1) && (c != 2) && (c != 3) && (c != 4) && (c != 5) && (c != 6) && (c != 0)) {
161             cout << "Error!" << endl;
162         }
163         if (c == 1) {
164             fill_table(table);
165         }
166         if (c == 2) {
167             table.show();
168         }
169         if (c == 3) {
170             string key;
171             key = input_str();
172             table.add(key);
173         }
174         if (c == 4) {
175             string key;
176             key = input_str();
177             table.del(key);
178         }
179         if (c == 5) {
180             string key;
181             key = input_str();
182             cout << table.find(key);
183         }
184         if (c == 6) {
185             w_file(table);
186         }
187     }
188     cout << "Thanks!" << endl;
189     return 0;
190 }

```

**Результаты хеш-функции:**





## Вывод

Изучил методы хеширования данных и получил практические навыки реализации хеш-таблиц. Как итог написал программу, характеристики которой соответствуют поставленному заданию.