

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**Пермский национальный исследовательский
политехнический университет**

Факультет электротехнический

Кафедра ИТАС

ОТЧЁТ

о лабораторной работе №18

Выполнил:
Студент группы ИВТ-23-1Б
Пискунов Д. А.

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Задача:

Написать программу строящую хеш-таблицу

Текст программы

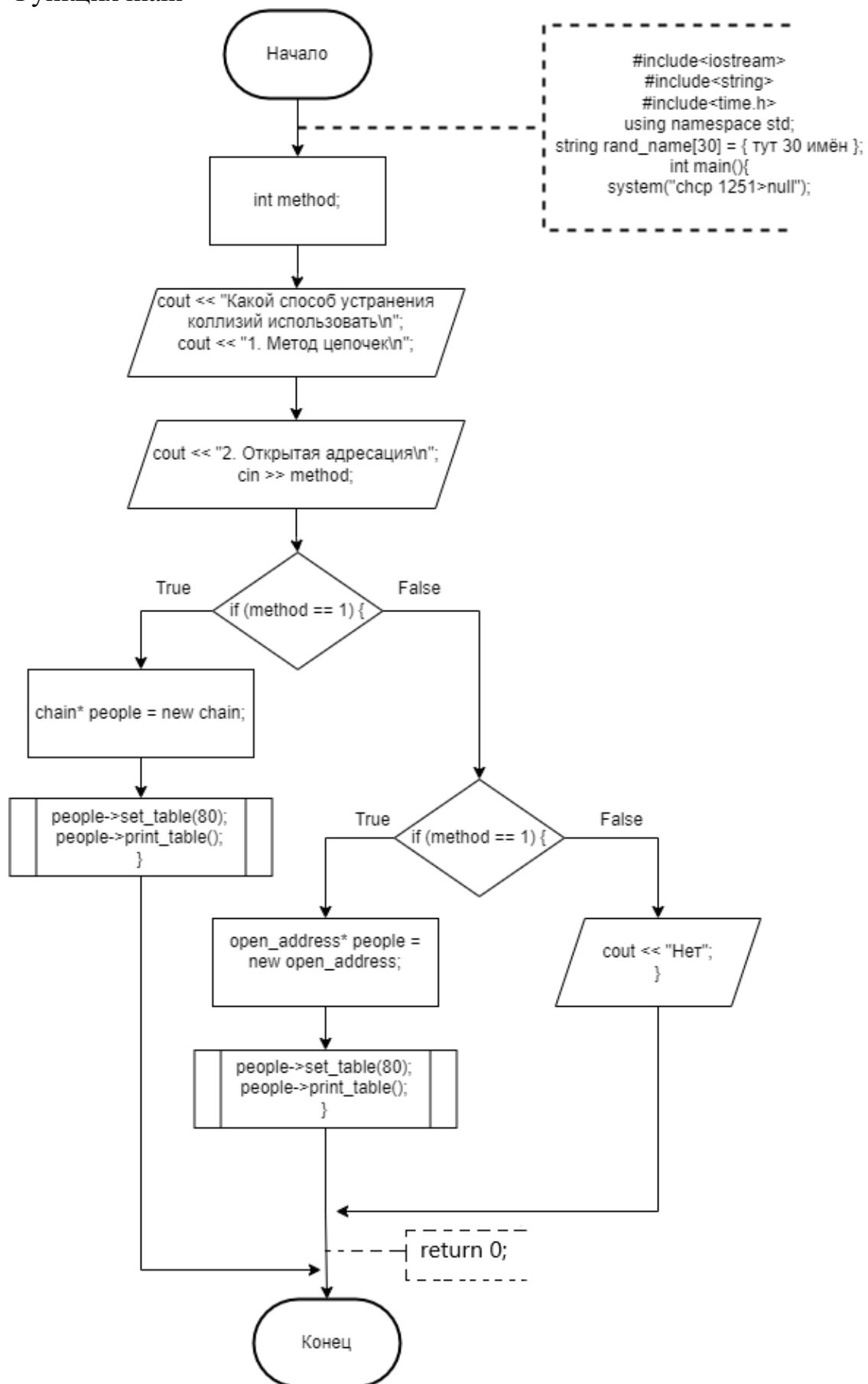
```
1  #include<iostream>
2  #include<string>
3
4  #include<time.h>
5
6  using namespace std;
7
8  string rand_name[30] = { "Иван", "Андрей", "Антон", "Амин", "Роберт", "Георгий", "Борис", "Владислав", "Дмитрий", "Никита", "Даниил", "Александр", "Сергей", "Михаил", "Илья", "Артём", "Максим", "Евгений", "Пётр", "Анатолий", "Константин", "Василий", "Андрей", "Антон", "Амин", "Роберт", "Георгий", "Борис", "Владислав", "Дмитрий", "Никита", "Даниил", "Александр", "Сергей", "Михаил", "Илья", "Артём", "Максим", "Евгений", "Пётр", "Анатолий", "Константин", "Василий" };
9
10
11 int get_hash(int size, string str) {
12     unsigned key = 0;
13     for (int i = 0; i < str.size(); i++) {
14         key += str[i];
15     }
16     return static_cast<int>(key * 0.2) % size;
17 }
18
19 struct node {
20     string name = " ";
21     int series;
22     int number;
23     int node_size = 0;
24     node* next = nullptr;
25     node* prev = nullptr;
26 };
27
28 struct open_address {
29     const int size = 50;
30     node* table = new node[size];
31
32     void set_node(string name, int series, int number) {
33         int hash = get_hash(size, name);
34         while (table[hash % size].name != " ") {
35             if (++hash - size == size) {
36                 break;
37             }
38         }
39         if (hash - size < size) {
40             table[hash % size].name = name;
41             table[hash % size].series = series;
42             table[hash % size].number = number;
43         }
44     }
45     void print_table() {
46         for (int i = 0; i < size; i++) {
47             if (table[i].name != " ") {
48                 cout << "Name: " << table[i].name << endl;
49                 cout << "Series: " << table[i].series << endl;
50                 cout << "Number: " << table[i].number << endl;
51                 cout << "=====\n";
52             }
53         }
54     }
55     void set_table(int count) {
56         srand(time(0));
57     }
58 }
```

```

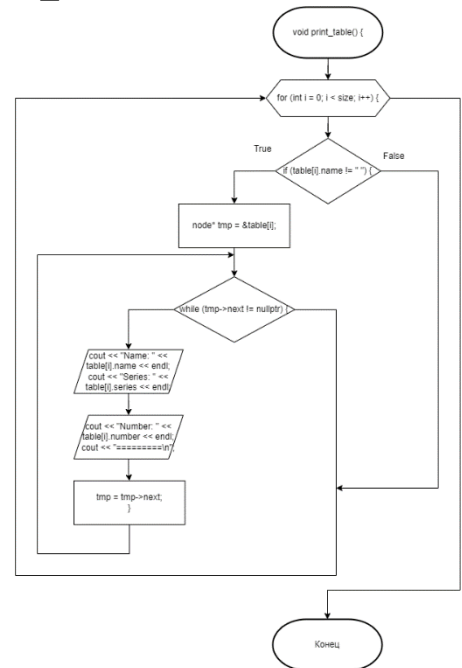
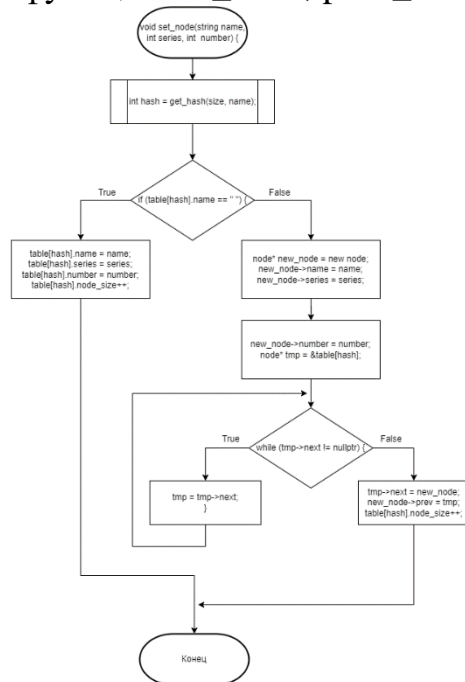
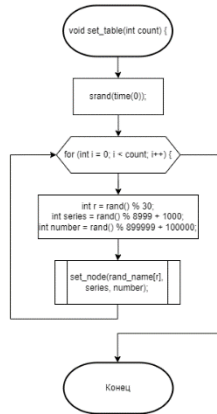
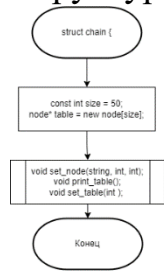
55 void set_table(int count) {
56     srand(time(0));
57     for (int i = 0; i < count; i++) {
58         int r = rand() % 30;
59         int series = rand() % 8999 + 1000;
60         int number = rand() % 899999 + 100000;
61         set_node(rand_name[r], series, number);
62     }
63 }
64 };
65
66 struct chain {
67     const int size = 50;
68     node* table = new node[size];
69
70     void set_node(string name, int series, int number) {
71         int hash = get_hash(size, name);
72         if (table[hash].name == " ") {
73             table[hash].name = name;
74             table[hash].series = series;
75             table[hash].number = number;
76             table[hash].node_size++;
77         }
78         else {
79             node* new_node = new node;
80             new_node->name = name;
81             new_node->series = series;
82             new_node->number = number;
83             node* tmp = &table[hash];
84             while (tmp->next != nullptr) {
85                 tmp = tmp->next;
86             }
87             tmp->next = new_node;
88             new_node->prev = tmp;
89             table[hash].node_size++;
90         }
91     }
92
93     void print_table() {
94         for (int i = 0; i < size; i++) {
95             if (table[i].name != " ") {
96                 node* tmp = &table[i];
97                 while (tmp->next != nullptr) {
98                     cout << "Name: " << table[i].name << endl;
99                     cout << "Series: " << table[i].series << endl;
100                     cout << "Number: " << table[i].number << endl;
101                     cout << "=====\n";
102                     tmp = tmp->next;
103                 }
104             }
105         }
106     }
107
108     void set_table(int count) {
109         srand(time(0));
110         for (int i = 0; i < count; i++) {
111             int r = rand() % 30;
112             int series = rand() % 8999 + 1000;
113             int number = rand() % 899999 + 100000;
114             set_node(rand_name[r], series, number);
115         }
116     }
117 };
118
119 int main() {
120     system("chcp 1251>0");
121
122     int method;
123
124     cout << "Какой способ устранения коллизий использовать\n";
125     cout << "1. Метод цепочек\n";
126     cout << "2. Открытая адресация\n";
127     cin >> method;
128
129     if (method == 1) {
130         chain* people = new chain;
131         people->set_table(80);
132         people->print_table();
133     }
134     else if (method == 2) {
135         open_address* people = new open_address;
136         people->set_table(80);
137         people->print_table();
138     }
139     else {
140         cout << "Нет";
141     }
142
143     return 0;
144 }
145
146

```

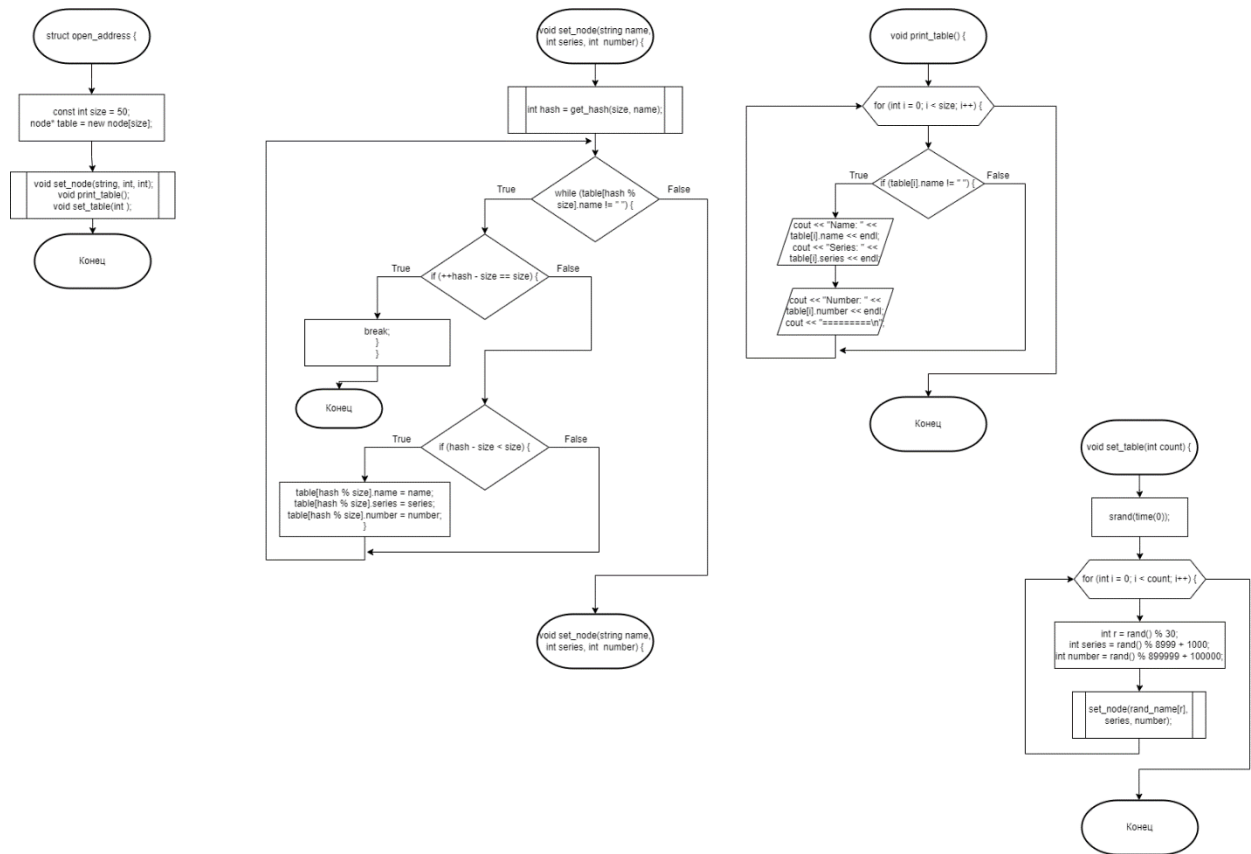
Блок схема
Функция main



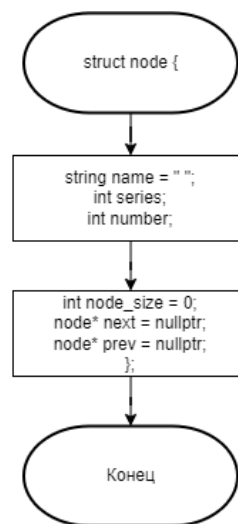
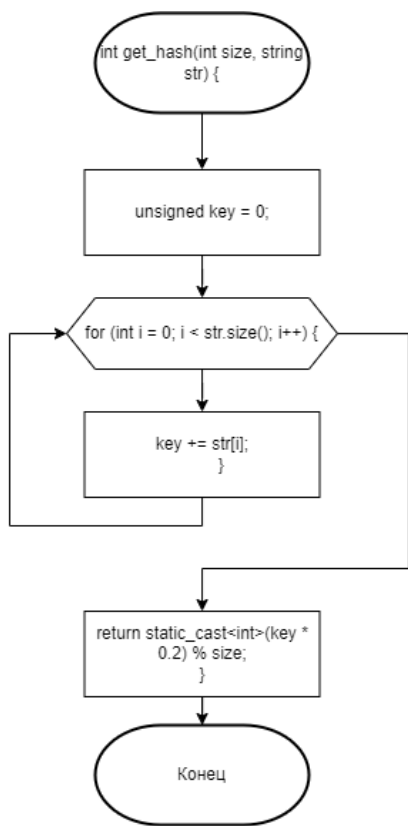
Структура chain и функции set_node, print_table, set_table



Структура open_address и функции set_node, print_table, set_table



Структура node и функция get_hash



Тесты

Какой способ устранения коллизий использовать

1. Метод цепочек
2. Открытая адресация

1

Name: Александр

Series: 2062

Number: 119420

=====

Name: Станислав

Series: 3222

Number: 126495

=====

Name: Станислав

Series: 3222

Number: 126495

=====

Name: Станислав

Series: 3222

Number: 126495

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Константин

Series: 4587

Number: 104603

=====

Name: Григорий

Series: 5568

Number: 128509

=====

Name: Валерий

Series: 5155

Number: 103273

=====

Name: Алексей

Series: 6069

Number: 129437

=====

Name: Богдан

Series: 2657

Number: 131550

```
=====
Name: Артем
Series: 7850
Number: 105357
=====
Name: Артем
Series: 7850
Number: 105357
=====
Name: Артем
Series: 7850
Number: 105357
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Антон
Series: 7614
Number: 103012
=====
Name: Амин
Series: 6769
Number: 101831
=====
Name: Артур
Series: 3820
Number: 114567
=====
```

C:\Users\MOkASiH\Desktop\Test\x64\Debug\Test.exe (процесс 16444) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...

Какой способ устранения коллизий использовать

1. Метод цепочек
2. Открытая адресация

2

Name: Николай

Series: 3565

Number: 120876

=====

Name: Сергей

Series: 8690

Number: 131839

=====

Name: Дамир

Series: 3324

Number: 100345

=====

Name: Дамир

Series: 5576

Number: 105484

=====

Name: Сурман

Series: 3496

Number: 128849

=====

Name: Георгий

Series: 3481

Number: 131719

=====

Name: Владислав

Series: 3457

Number: 112807

=====

Name: Григорий

Series: 5478

Number: 117727

=====

Name: Борис

Series: 4267

Number: 110520

=====

Name: Антон

Series: 9115

Number: 127147

=====

Name: Денис

Series: 6717

Number: 123178

=====

Name: Иннокентий

Series: 5859

Number: 104782

=====

Name: Станислав

Series: 2966

Number: 125668

=====

Name: Илья

Series: 3997

Number: 113588

=====

Name: Богдан

Series: 5909

Number: 105092

Name: Артур
Series: 7918
Number: 111761
=====

Name: Иван
Series: 9028
Number: 108679
=====

Name: Дамир
Series: 4394
Number: 112317
=====

Name: Роберт
Series: 6396
Number: 104856
=====

Name: Артем
Series: 2025
Number: 116193
=====

Name: Сурман
Series: 9483
Number: 124039
=====

Name: Иван
Series: 3638
Number: 106424
=====

Name: Денис
Series: 6759
Number: 118147
=====

Name: Артем
Series: 5156
Number: 124697
=====

Name: Данила
Series: 6609
Number: 111057
=====

Name: Артем
Series: 6626
Number: 110112
=====

Name: Герман
Series: 4999
Number: 101374
=====

Name: Михаил
Series: 9899
Number: 117612
=====

Name: Антон
Series: 9196
Number: 121852
=====

Name: Илья
Series: 5570
Number: 112814
=====

C:\Users\MOKASIN\Desktop\Test\x64\Debug\Test.exe (процесс 676) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно...