

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 №11 (двунаправленный список)

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

Проверил:
Доцент кафедры ИТАС
Яруллин Д.В.

Пермь 2024

Постановка задач

1. Создание стека.
2. Добавление элемента в список (в соответствии со своим вариантом).
3. Удаление элемента из списка (в соответствии со своим вариантом).
4. Печать стека.
5. Запись стека в файл.
6. Уничтожение стека.
7. Восстановление стека из файла.

Вариант 20

Записи в линейном списке содержат ключевое поле типа `*char` (строка символов). Сформировать стек. Удалить элемент с заданным ключом. Добавить по *K* элементов в начало и в конец списка.

Текст программы

```

1  #include<iostream>
2  #include<string>
3  #include<fstream>
4
5  using namespace std;
6
7
8  struct StackElem {
9      char data = 0;
10     StackElem* downner = nullptr;
11 };
12 struct Stack{
13     StackElem* top = nullptr;
14     int size = 0;
15     void push(char data) {
16         StackElem* newElem = new StackElem;
17         newElem->data = data;
18         newElem->downner = top;
19         top = newElem;
20         size++;
21     }
22     char pop() {
23         if (size == 0) {
24             cout << "Стек пуст, действие невозможно" << endl;
25             return NULL;
26         }
27         char value = top->data;
28         StackElem* tmp = top;
29         top = top->downner;
30         size--;
31         delete tmp;
32         return value;
33     }
34 }s;
35
36
37 void print(Stack& s) {
38     if (s.size == 0) {
39         cout << "Стек пуст." << endl;
40         return;
41     }
42     Stack tmp;
43     char datatmp, sized = s.size;
44     for (int i = 0; i < sized; i++) {
45         datatmp = s.pop();
46         tmp.push(datatmp);
47     }
48     for (int i = 0; i < sized; i++) {
49         datatmp = tmp.pop();
50         s.push(datatmp);
51         cout << datatmp;
52     }
53 }
54
55 void add(Stack& s, char data, int n) {
56     Stack tmp;
57     char datatmp;
58     int sized = s.size;
59     for (int i = 0; i <= sized - n; i++) {
60         datatmp = s.pop();
61         tmp.push(datatmp);
62     }
63     s.push(data);
64     for (int i = 0; i <= sized - n; i++) {
65         datatmp = tmp.pop();
66         s.push(datatmp);
67     }
68 }
69
70 void del(Stack& s, int n) {

```

```

70 void del(Stack& s, int n) {
71     Stack tmp;
72     char datatmp;
73     int sized = s.size;
74     for (int i = 0; i < sized - n; i++) {
75         datatmp = s.pop();
76         tmp.push(datatmp);
77     }
78     s.pop();
79     for (int i = 0; i < sized - n; i++) {
80         datatmp = tmp.pop();
81         s.push(datatmp);
82     }
83 }
84
85 bool search(Stack& s, char da) {
86     Stack tmp;
87     char datatmp;
88     bool flag = false;
89     int size = s.size;
90     for (int i = 0; i < size; i++) {
91         datatmp = s.pop();
92         tmp.push(datatmp);
93     }
94     for (int i = 0; i < size; i++) {
95         datatmp = tmp.pop();
96         s.push(datatmp);
97         if (datatmp == da) {
98             flag = true;
99         }
100     }
101     return flag;
102 }
103
104 void saver(Stack& s) {
105     ofstream file1("C:\\Users\\MOKASIH\\Desktop\\sen_2\\11\\save.txt");
106     if (file1) {
107         Stack tmp;
108         char datatmp;
109         int size = s.size;
110         for (int i = 0; i < size; i++) {
111             datatmp = s.pop();
112             tmp.push(datatmp);
113         }
114         for (int i = 0; i < size; i++) {
115             datatmp = tmp.pop();
116             s.push(datatmp);
117             file1 << datatmp;
118         }
119     }
120     else {
121         cout << "Файл не открывается";
122     }
123     file1.close();
124 }
125
126 void killer(Stack& s) {
127     int size = s.size;
128     for (int i = 0; i < size; i++) {
129         s.pop();
130     }
131 }
132
133 void load(Stack& s) {
134     ifstream file1("C:\\Users\\MOKASIH\\Desktop\\sen_2\\11\\save.txt");
135     if (file1) {
136         cin.ignore();
137         string f;
138         getline(file1, f);
139     }

```

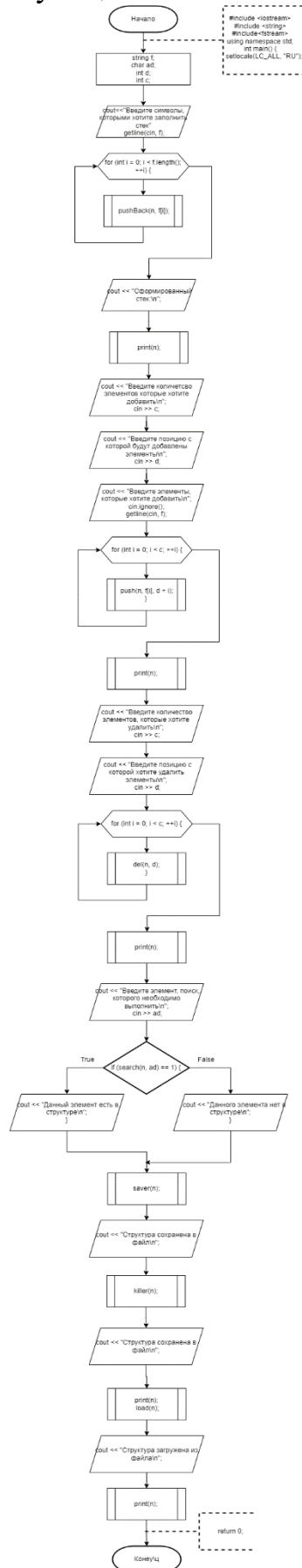
```

139     getline(file1, f);
140     for (int i = 0; i < f.length(); i++) {
141         s.push(f[i]);
142     }
143     file1.close();
144 }
145 else {
146     cout << "Файл не открыт";
147 }
148 }
149
150
151
152
153 int main() {
154     setlocale(LC_ALL, "RU");
155     int p, d;
156     char ad;
157     string f;
158     cout << "Введите символы которыми хотите заполнить стек\n";
159     getline(cin, f);
160     for (int i = 0; i < f.length(); i++)
161     {
162         s.push(f[i]);
163     }
164
165     cout << "Сформированный стек\n";
166     print(s);
167
168     cout << "\nВведите количество элементов которые хотите добавить\n";
169     cin >> d;
170     cout << "Введите позиции на которую хотите добавить элементы\n";
171     cin >> p;
172     cout << "Введите элементы которые хотите добавить\n";
173     cin.ignore();
174     getline(cin, f);
175     for (int i = 0; i < d; i++) {
176         add(s, f[i], p + i);
177     }
178     print(s);
179
180     cout << "\nВведите количество элементов которые хотите удалить\n";
181     cin >> d;
182     cout << "Введите позиции с которой хотите удалить элементы\n";
183     cin >> p;
184     for (int i = 0; i < d; i++) {
185         del(s, p);
186     }
187     print(s);
188
189
190     cout << "\nВведите элемент, поиск, которого необходимо выполнить\n";
191     cin >> ad;
192     if (search(s, ad) == 1) {
193         cout << "Данный элемент есть в структуре\n";
194     }
195     else {
196         cout << "Данного элемента нет в структуре\n";
197     }
198
199     saver(s);
200     cout << "Структура сохранена в файл\n";
201     killer(s);
202     cout << "Структура удалена\n";
203     print(s);
204
205     load(s);
206     cout << "Структура загружена из файла\n";
207     print(s);
208
209     print(s);
210     return 0;
211 }

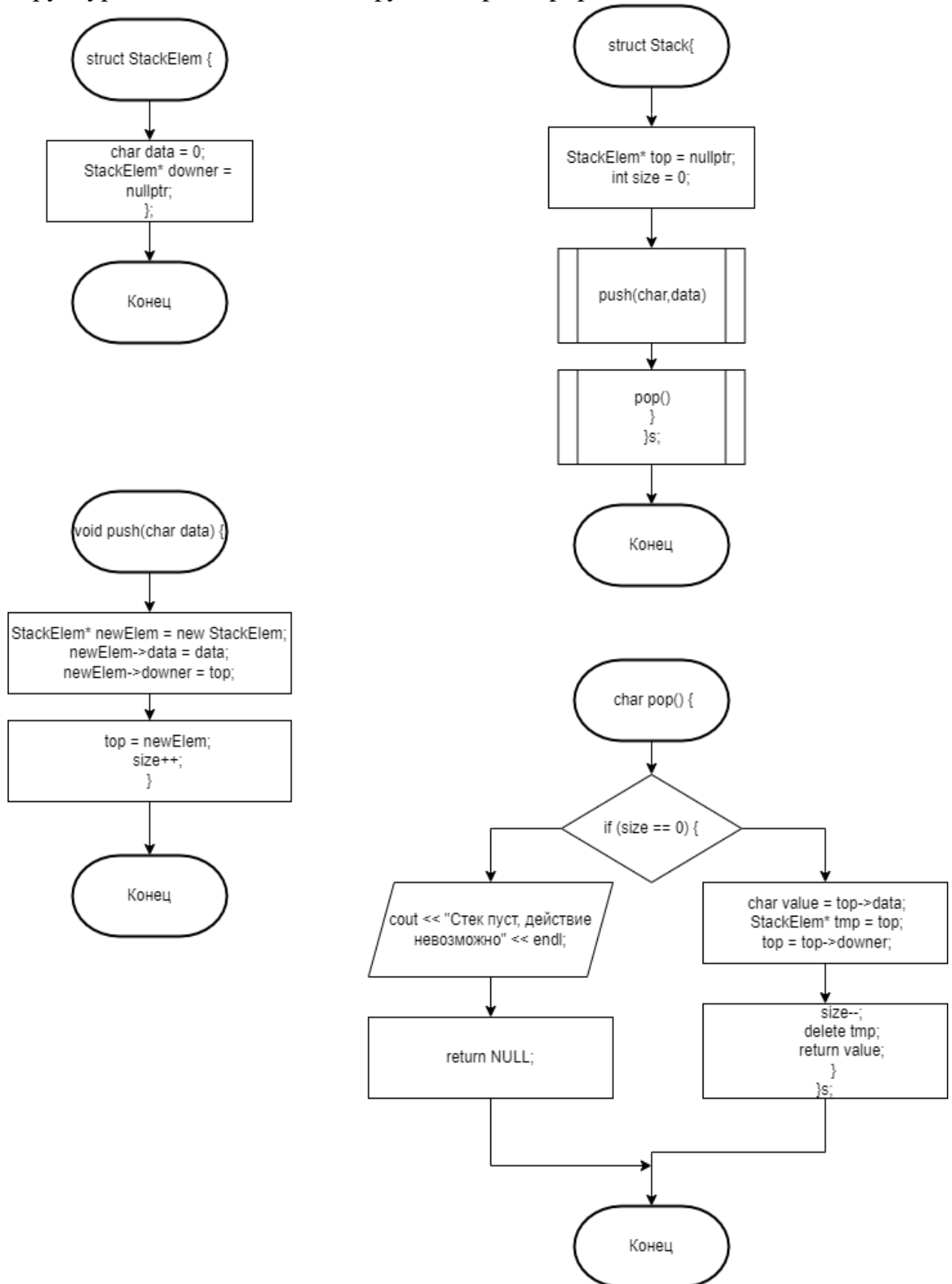
```

3) Блок схема

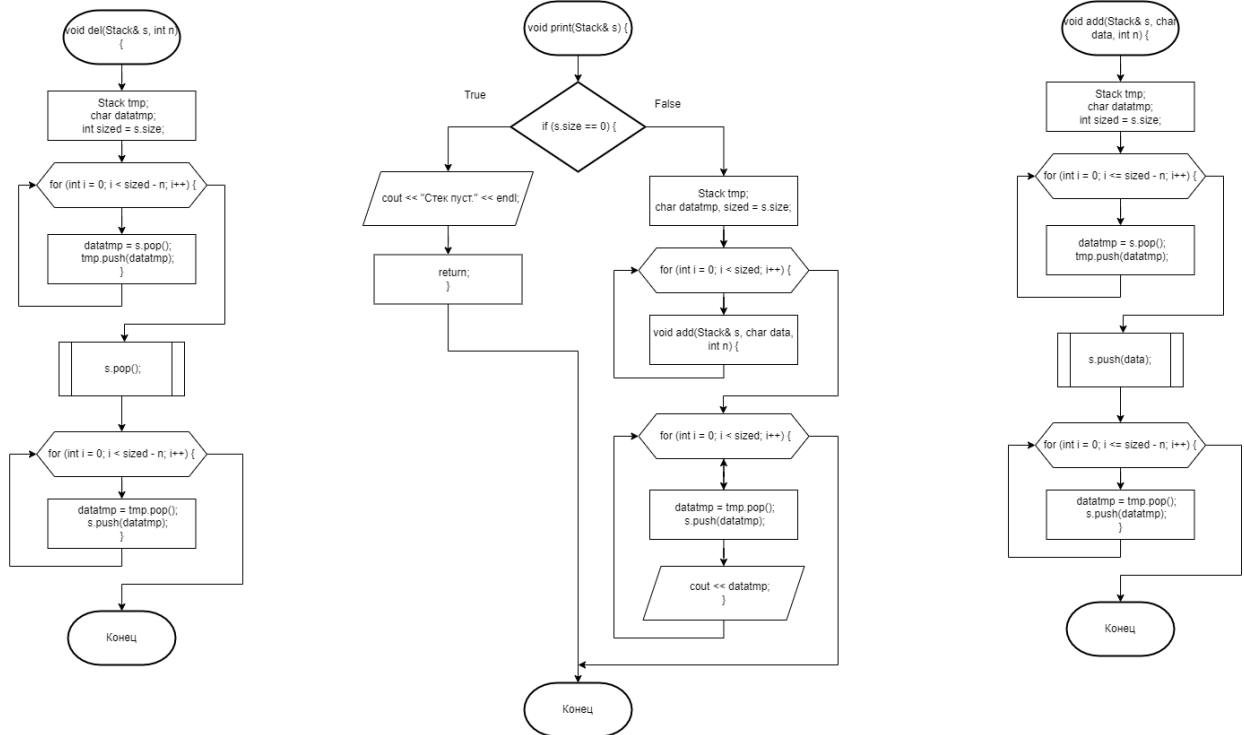
Функция main



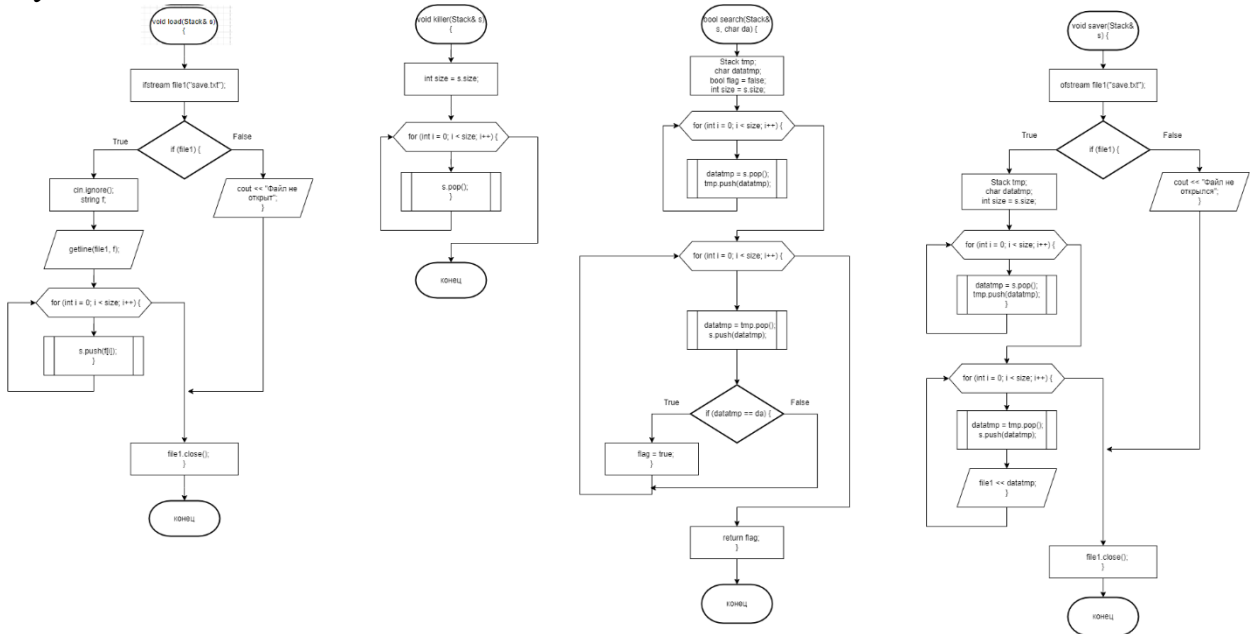
структуры StackElem, Stack и функции push, pop



Функции print, del, add



Функции search, saver, killer, load



Тест

```
Введите символы которыми хотите заполнить стек
goodbye
Сформированный стек
goodbye
Введите количество элементов которые хотите добавить
7
Введите позицию на которую хотите добавить элементы
9
Введите элементы которые хотите добавить
america
goodbye americ
Введите количество элементов которые хотите удалить
8
Введите позицию с которой хотите удалить элементы
1
americ
Введите элемент, поиск, которого необходимо выполнить
a
Данный элемент есть в структуре
Структура сохранена в файл
Структура удалена
Стек пуст.
Структура загружена из файла
americ
C:\Users\MOKASIH\Desktop\labs\x64\Debug\labs.exe (процесс 7436) завершил работу с кодом 0.
Нажмите любую клавишу, чтобы закрыть это окно: █
```