Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе №11**

Дисциплина: основы алгоритмизации и программирования

Тема: “ Динамические структуры данных.

Двунаправленные списки”

Вариант 16

Выполнил:

студент группы ИВТ-20-2Б Тедеев А.З.

Проверила: доцент кафедры ИТАС

Полякова О.А.

Пермь, 2021

**Цель задачи**

Получить практические навыки работы с двунаправленными списками.

**Постановка задачи**

Тип информационного поля int. Удалить из списка первый элемент с четным информационным полем.

2

**Анализ задачи**

**1.** Для решения задачи необходимо:

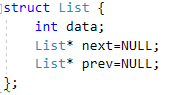
**1.1.** Организовать структуру, поля которой будут использоваться для ввода данных типа int, для хранения адреса следующего и предыдущего элемента.

**1.2.** Организовать функцию make() для формирования списка.

**1.3.** Организовать функцию Deletion() для удаления элемента с чётным номером.

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Структура List для организации двунаправленного списка.



**2.2.** Для функции make() используется в качестве параметра int size - количество элементов в списке.

List\* make(int size)

**2.3.** Для функции Deletion() используется указатель на первый элемент в списке в качестве параметра

List\* &first

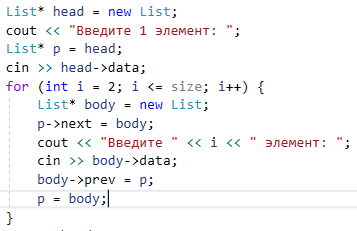
**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Данные вводятся через консоль в виде двунаправленного списка через структуру (тип List и полями типа int).

**4.** Для операций ввода и вывода использовались следующие операторы и функции:

**4.1.** Ввод данных на консоль реализован функцией cin для первого элемента списка и в цикле for (также через cin) для следующих.

3



**4.2.** Переменная n, отвечающая за размерность списка, проверяется в цикле while, чтобы не ввести отрицательное число.

while (size <1) {

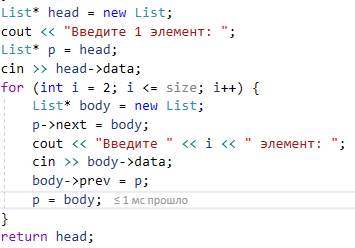
cout << "Введите размер списка: ";

cin >> size;

}

**5.** Поставленные задачи будут решены следующими действиями:

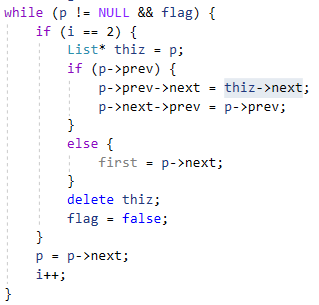
**5.1.** В функции make() осуществляется ввод всех элементов списка. Первый вводится вне цикла for и используется в качестве указателя на список. В цикле for заполняются элементы от 2 до size: резервируется память в переменной body, в которую мы вводим следующий элемент после первого, а первый элемент сохраняется в методе body.prev, а следующий после него в body.next.



**5.2.** В функции Deletion() используются переменная I в качестве итератора, flag типа bool, которая будет принимать значение false после удаления элемента. Когда i станет равной 2, производим удаление путем изменения связей между первым и третьим элементами, то есть, когда в цикле указывается второй элемент,

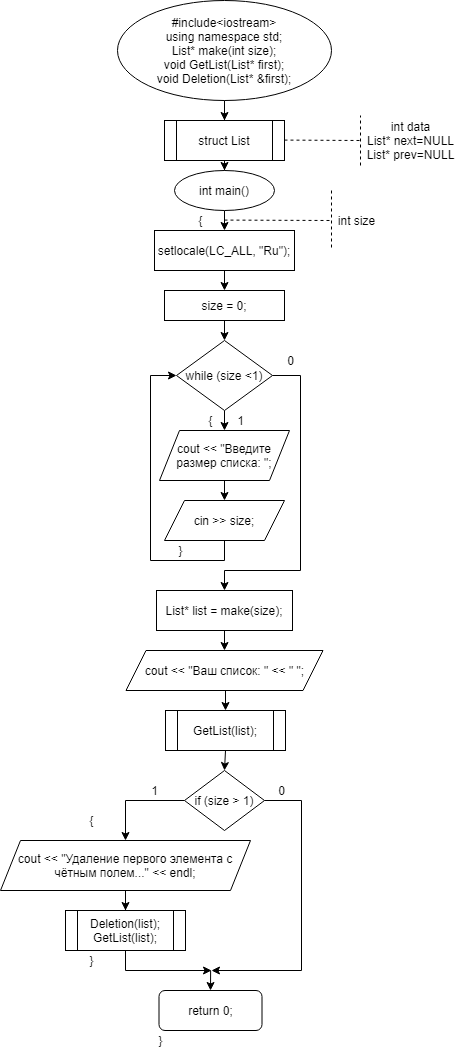
4

вместо него необходимо указать третий элемент через метод p.prev.next (p.prev – указатель на первый элемент, p.prev.next указатель на второй элемент, который нужно подменить) thiz->next (указатель на третий элемент), а для третьего элемента сделать первый предыдущим.

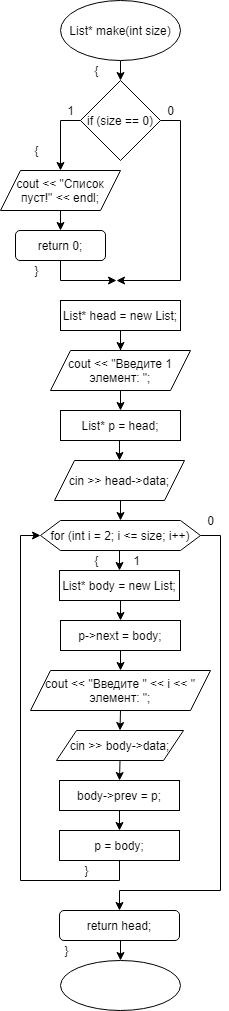


5

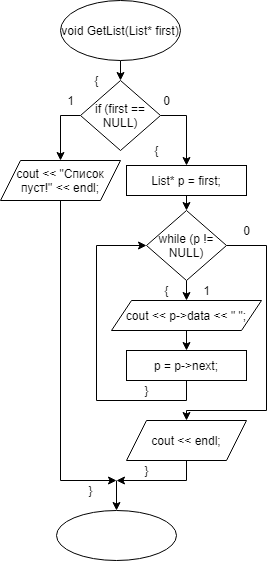
**Блок-схема**



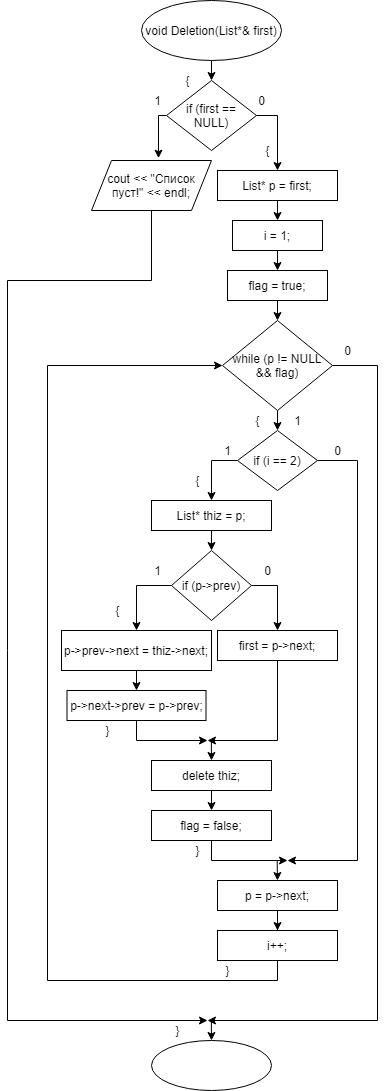
6



7

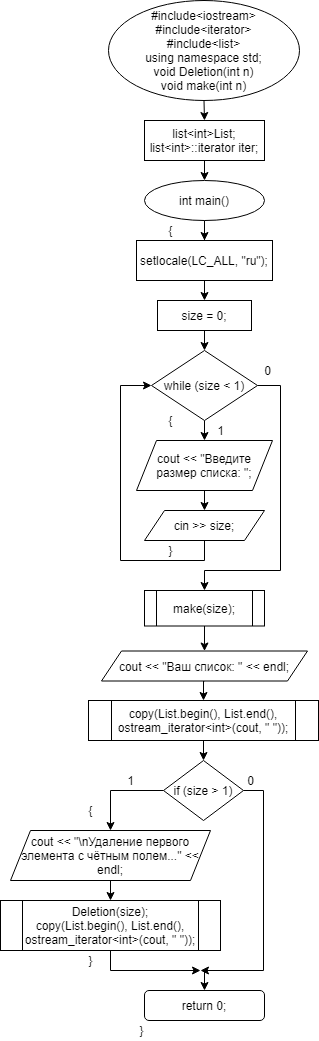


8

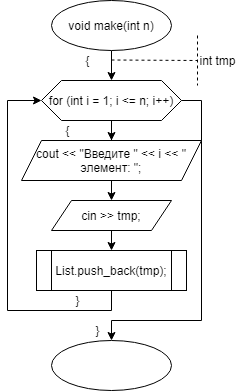


9

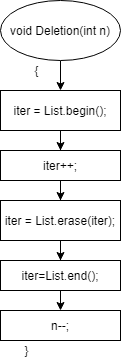
**Реализация через STL библиотеки**



10



11



12

**Код**

**Реализация через структуры**

#include<iostream>

using namespace std;

struct List {

int data;

List\* next=NULL;

List\* prev=NULL;

};

List\* make(int size);

void GetList(List\* first);

void Deletion(List\* &first);

int main()

{

setlocale(LC\_ALL, "Ru");

int size = 0;

while (size <1) {

cout << "Введите размер списка: ";

cin >> size;

}

List\* list = make(size);

cout << "Ваш список: " << " ";

GetList(list);

if (size > 1) {

cout << "Удаление первого элемента с чётным полем..." << endl;

Deletion(list);

GetList(list);

}

}

List\* make(int size)

{

if (size == 0) {

cout << "Список пуст!" << endl;

return 0;

}

List\* head = new List;

cout << "Введите 1 элемент: ";

List\* p = head;

cin >> head->data;

for (int i = 2; i <= size; i++) {

List\* body = new List;

p->next = body;

cout << "Введите " << i << " элемент: ";

cin >> body->data;

body->prev = p;

p = body;

}

return head;

}

void GetList(List\* first)

{

if (first == NULL)

cout << "Список пуст!" << endl;

else {

List\* p = first;

while (p != NULL) {

cout << p->data << " ";

p = p->next;

}

13

cout << endl;

}

}

void Deletion(List\*& first)

{

if (first == NULL)

cout << "Список пуст!" << endl;

else {

List\* p = first;

int i = 1;

bool flag = true;

while (p != NULL && flag) {

if (i == 2) {

List\* thiz = p;

if (p->prev) {

p->prev->next = thiz->next;

p->next->prev = p->prev;

}

else {

first = p->next;

}

delete thiz;

flag = false;

}

p = p->next;

i++;

}

}

}

14

**Реализация через структуры**

#include<iostream>

#include<iterator>

#include<list>

using namespace std;

list<int>List;

list<int>::iterator iter;

void make(int n) {

int tmp;

for (int i = 1; i <= n; i++) {

cout << "Введите " << i << " элемент: ";

cin >> tmp;

List.push\_back(tmp);

}

}

void Deletion(int n) {

iter = List.begin();

iter++;

iter = List.erase(iter);

iter=List.end();

n--;

}

int main()

{

setlocale(LC\_ALL, "ru");

int size = 0;

while (size < 1) {

cout << "Введите размер списка: ";

cin >> size;

}

make(size);

cout << "Ваш список: " << endl;

copy(List.begin(), List.end(), ostream\_iterator<int>(cout, " "));

if (size > 1) {

cout << "\nУдаление первого элемента с чётным полем..." << endl;

Deletion(size);

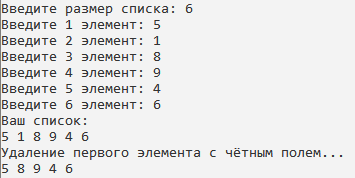
copy(List.begin(), List.end(), ostream\_iterator<int>(cout, " "));

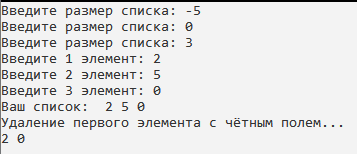
}

}

15

**Скриншоты**





16