vpМинистерство 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: "Информационные динамические структуры" в C++

Выполнил работу

студент группы ИВТ-20-1б

Кущ Д.В.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

Цель – Знакомство с динамическими информационными

структурами на примере одно- и двунаправленных списков.

Для достижения поставленной цели необходимо решить следующие задачи:

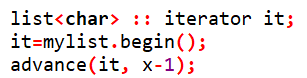
* Найти как создаются и объявляются списки данных.
* Найти как работают списки данных.
* Разработать программу по найденному материалу.

Постановка задачи

Задача состоит в написании программы в двух вариантах (через библиотеки STL и через структуру данных), с созданием двунаправленного списка с типом данных “char”, заполнением списка данными, удалением элементов равных запрашиваемому элементу, а также добавлением некого количества элементов, введенных пользователем, перед указанным элементом списка. Для выполнения данной работы использован язык программирования C++ в программной среде DEV-C++.

Анализ задачи

Вариант через библиотеки STL.

1. Подключаем необходимые библиотеки такие как библиотека двунаправленных списков и библиотека итераторов:  
   
2. Создание списка:  
   
3. Ввод данных в список производится через цикл for, с параметром, увеличивающимся до размера списка, запрашивая элемент у пользователя и вводя его при помощи функции:  
   
4. Вывод списка происходит при помощи функции:  
   
5. Для удаления мне необходимо запросить элемент у пользователя, а затем при помощи функции удалить все схожие с ним элементы:  
     
   Где ‘k’ это запрашиваемый у пользователей элемент.
6. Для добавления элементов в список я запрашиваю у пользователя количество добавляемых элементов, запрашиваю номер элемента, перед которым необходимо добавить элементы. Затем создаю переменную указателя, приравниваю ее к указателю первого элемента и сдвигаю указатель до нужного элемента:  
     
   Далее создаю новый список, в который вношу данные, которые пользователь хочет добавить в список и вставляю список в указанное место при помощи функции:  
   
7. Так же после каждой функции (т.е. удаления и добавления элементов), выводим новый измененный список при помощи функции, описанной под цифрой 2.
8. Также при составлении программы я учитываю, что пользователь будет вводить размер списка >0, а также будет вводить нужные типы данных.

**Код**

#include <iostream>

#include <list>

#include <iterator>

using namespace std;

list<char> mylist;

void make(int n){

char tmp;

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

cout << "Введите " << i+1 << " элемент: ";

cin >> tmp;

mylist.push\_back(tmp);

}

}

void del() {

char k;

cout << "Введите элемент, который необходимо удалить: " << endl;

cin >> k;

mylist.remove(k);

}

void add() {

int k;

cout << "Введите кол-во добавляемых элементов: " << endl;

cin >> k;

int x;

cout << "Введите номер элемента перед которым необходимо добавить" << endl;

cin >> x;

list<char> :: iterator it;

it=mylist.begin();

advance(it, x-1);

char tmp;

list<char> additlist;

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

cout << "Введите " << i+1 << " новый элемент: ";

cin >> tmp;

additlist.push\_back(tmp);

}

mylist.insert(it,additlist.begin(),additlist.end());

}

int main() {

setlocale(LC\_ALL, "Rus");

int size;

cout << "Введите размер списка" << endl;

cin >> size;

make(size);

cout << "Ваш список: " << endl;

copy(mylist.begin(), mylist.end(), ostream\_iterator<char>(cout, " "));

cout << endl;

del();

cout << "Список после удаления: " << endl;

copy(mylist.begin(), mylist.end(), ostream\_iterator<char>(cout, " "));

cout << endl;

add();

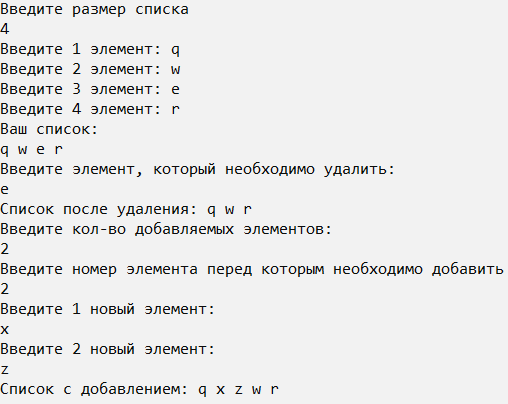
cout << "Cписок c добавлением: " << endl;

copy(mylist.begin(), mylist.end(), ostream\_iterator<char>(cout, " "));

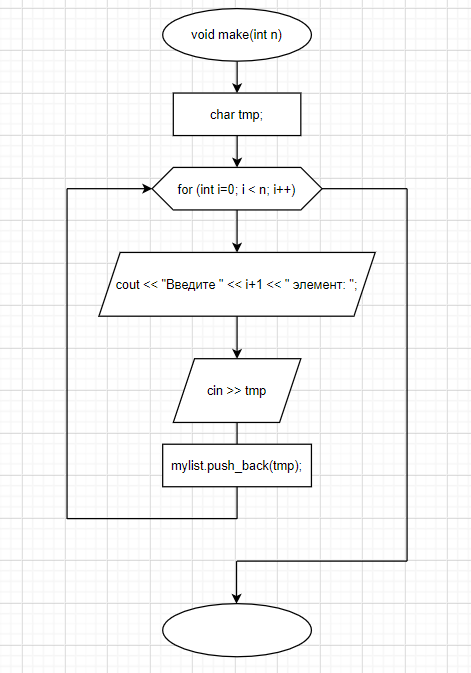
return 0;

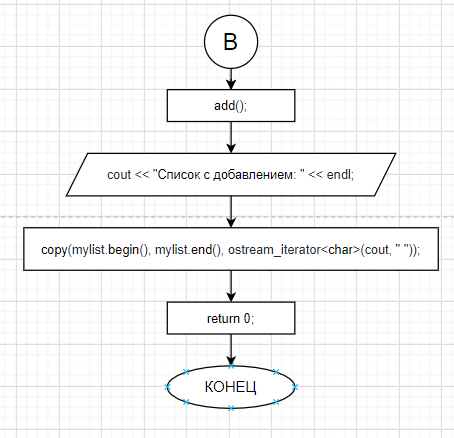
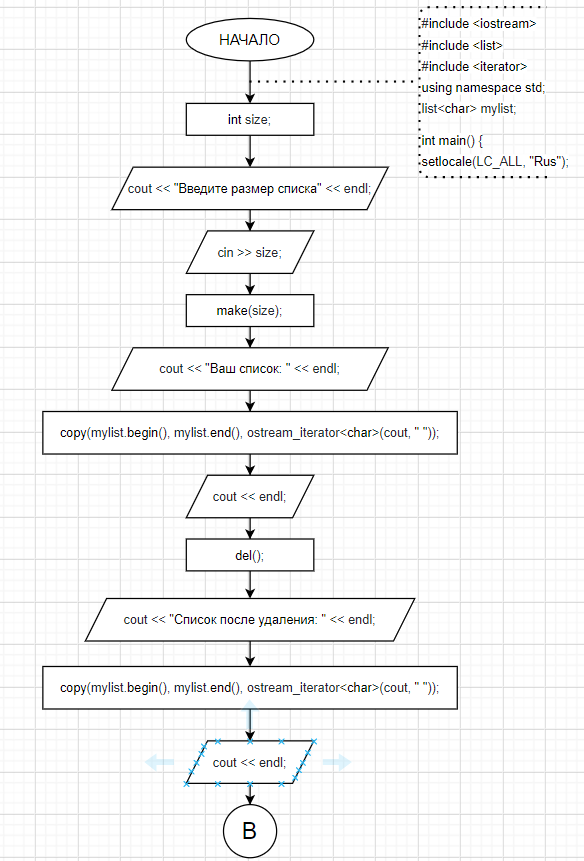
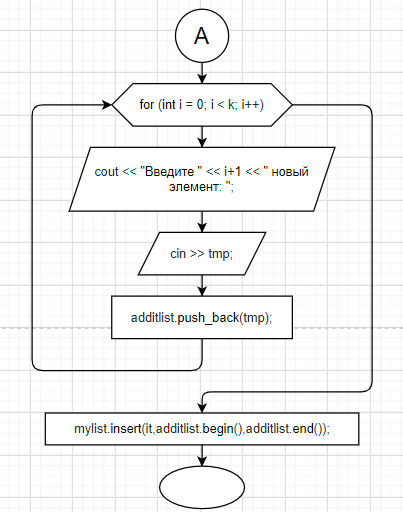
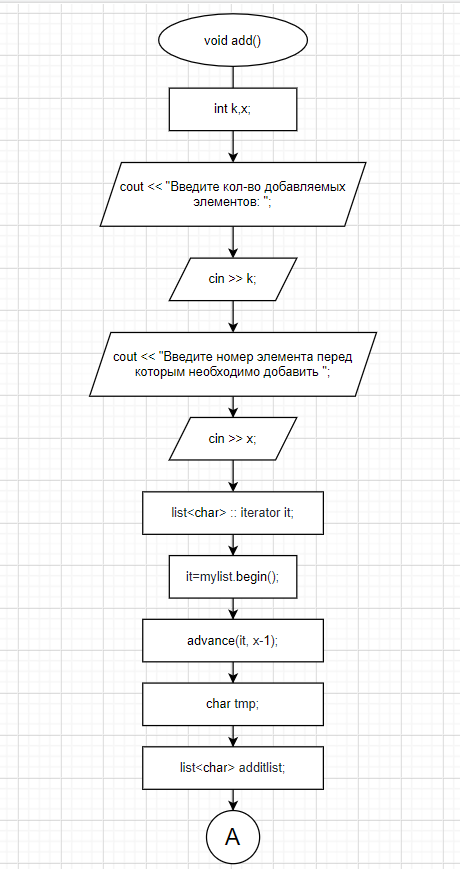
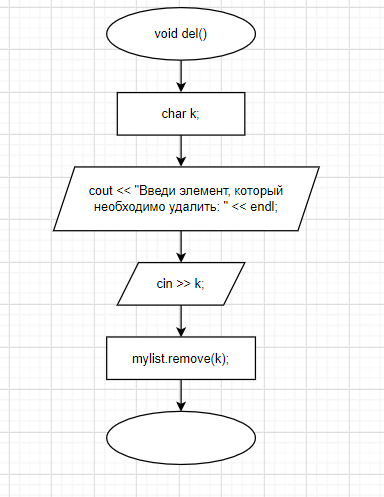
}

Пример вывода консоли

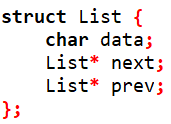


Блок схема





Вариант через структуру данных.

1. Создание структуры списка:  
   
2. Для ввода элементов в список я сначала ввожу только один, первый элемент, устанавливаю связь next на NULL и связь prev на NULL, т.к. у меня только один элемент на данный момент. Затем через цикл for начиная с 2 до размера списка вводим оставшиеся элементы и простраиваем их связь друг с другом. Возвращаем из функции указатель на первый элемент.
3. Вывод списка происходит через цикл while, выводя элементы и сдвигая указатель p вперед, пока он не будет указывать на NULL, т.е. мы не достигнем последнего элемента.
4. Для удаления элементов мы запрашиваем элемент, который необходимо удалить у пользователя, затем начиная с первого элемента проверяем все начальные элементы на равенство и в случае истины удаляем их. Затем т.к. мы прошли все элементы, равные нужному элементу в списке, следующий элемент не является подходящим. Мы пропускаем этот элемент, начиная цикл while со следующего элемента до тех пор пока не дойдем до конца списка, через условный оператор if проверяем равенство элемента, на который показывает указатель, и в случае истины удаляем его, перестраиваем связи и переходим на следующий элемент, если условие ложно, просто переходим на следующий элемент.
5. Для добавления элементов в список я запрашиваю у пользователя количество добавляемых элементов, запрашиваем номер элемента, перед которым необходимо добавить элементы. Затем создаем массив данных типа char с размером количества вводимых данных и вводим в него новые элементы запрашивая их у пользователя. Указателю p присваиваем значения указателя на первый элемент. Дальше через условный оператор проверяем надо ли нам добавить элементы в начало списка. Если условие истинно, то через цикл for, идущий от количества добавляемых элементов до 0, создаем новые элементы списка и вводим их значение из массива, созданного ранее. Для параметра массива используем значение параметра цикла for, используемого для перебора. При вводе нового элемента, простраиваем измененные связи списка.  
   Важно!  
   Я использовал дополнительный массив данных, т.к. при обычном вводе элементы заносятся в обратном порядке. Поэтому я ввел элементы в массив данных и брал их из него в обратном порядке (цикл for начинался с максимального элемента и шел до 0).
6. Так же после каждой функции (т.е. удаления и добавления элементов), выводим новый измененный список при помощи функции, описанной под цифрой 2.
7. Также при составлении программы я учитываю, что пользователь будет вводить размер списка >0, а также будет вводить нужные типы данных.

**Код**

#include <iostream>

using namespace std;

struct List {

char data;

List\* next;

List\* prev;

};

List\* make(int size) {

List\* first, \*p;

first = NULL;

p = new List;

cout << "Введите 1 элемент: ";

cin >> p->data;

first = p;

p->next = NULL;

p->prev = NULL;

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

List\* h = new List;

List\* q = p;

p->next = h;

p = p->next;

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

cin >> p->data;

p->next = NULL;

p->prev = q;

}

return first;

}

void print(List\*first) {

List\* p = first;

while (p != NULL) {

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

p = p->next;

}

cout << endl;

}

void del(List\* &first) {

List\* p = first;

char k;

cout << "Введите элемент, который необходимо удалить: " << endl;

cin >> k;

while(p->data == k) {

List\* q = first;

first = p->next;

delete q;

p = first;

p->prev = NULL;

}

while (p->next != NULL) {

if (p->next->data == k) {

List\* q = p->next;

List\* d = p->next->prev;

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

p->next->prev = d;

delete q;

}

else p = p->next;

}

}

void add(List\* &first, int n) {

int k;

cout << "Введите кол-во добавляемых элементов: " << endl;

cin >> k;

int x;

cout << "Введите номер элемента перед которым необходимо добавить" << endl;

cin >> x;

List\* p = first;

char arr[k];

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

cout << "Введите " << i+1 << " новый элемент: " << endl;

cin >> arr[i];

}

if(x==1){

for (int i = k-1; i >= 0; i--) {

List\* h = new List;

h->data=arr[i];

h->next = first;

first->prev = h;

h->prev = NULL;

first = h;

}

}else{

for(int i=1;i<x-1;i++){

p=p->next;

}

for (int i = k-1; i >= 0; i--) {

List\* h = new List;

h->data = arr[i];

h->next = p->next;

h->prev = p;

p->next->prev = h;

p->next = h;

}

}

}

int main() {

setlocale(LC\_ALL, "Rus");

int size;

cout << "Введите размер списка" << endl;

cin >> size;

List\* list = make(size);

cout << "Ваш список: " << endl;

print(list);

del(list);

cout << "Список после удаления: ";

print(list);

add(list,size);

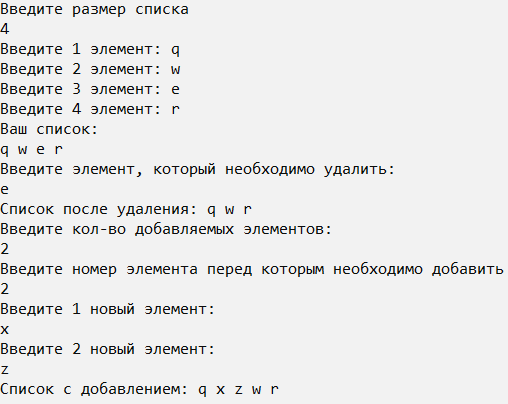
cout << "Cписок c добавлением: ";

print(list);

return 0;

}

Пример вывода консоли



Блок схема