Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Основы алгоритмизации и программирования»

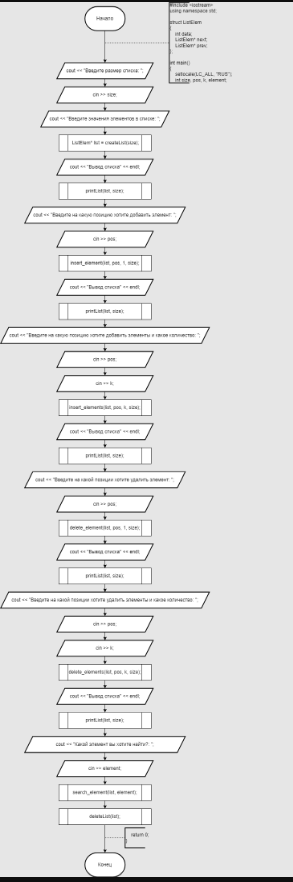
Тема: Лабораторная работа №11, Однонаправленный список

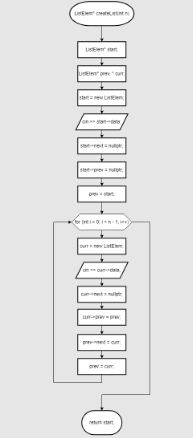
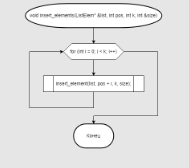
Выполнил работу  
Студент группы РИС-22-1Б  
Ишемцева Мария Андреевна  
Проверил  
Доцент кафедры ИТАС  
Полякова Ольга Андреевна

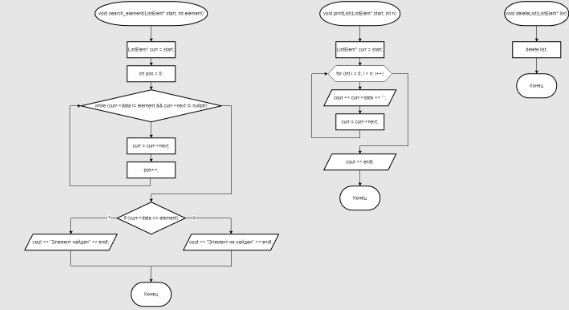
**Постановка задачи**

Реализовать однонаправленный список через структуры, добавление и удаление элементов

**Блок-схема**





**Программа**

#include <iostream>

using namespace std;

struct ListE

{

int data;

ListE\* next;

};

ListE\* createList(int n)

{

ListE\* start;

ListE\* prev, \* curr;

start = new ListE;

cin >> start->data;

start->next = nullptr;

prev = start;

for (int i = 0; i < n - 1; i++)

{

curr = new ListE;

cin >> curr->data;

curr->next = nullptr;

prev->next = curr;

prev = curr;

}

return start;

}

void insert\_element(ListE\*& start, int pos, int k, int& size)

{

size++;

ListE\* first = start;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;

}

ListE\* new\_element = new ListE;

cout << "Введите элемент: ";

cin >> new\_element->data;

if (pos == 1)

{

new\_element->next = first;

start = new\_element;

}

else

{

new\_element->next = first->next;

first->next = new\_element;

}

}

void insert\_elements(ListE\*& list, int pos, int k, int& size)

{

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

{

insert\_element(list, pos + i, k, size);

}

}

void delete\_element(ListE\*& start, int pos, int k, int& size)

{

size--;

ListE\* first = start;

ListE\* last;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;

}

if (pos == 1)

{

start = first->next;

first->next = start;

}

else

{

last = first->next->next;

first->next = last;

}

}

void delete\_elements(ListE\*& start, int pos, int k, int& size)

{

size -= k;

ListE\* first = start;

ListE\* last;

for (int i = 1; i < pos - 1; i++)

{

first = first->next;;

}

last = first;

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

{

last = last->next;

}

if (pos == 1)

{

start = last;

}

else

{

first->next = last->next;

}

}

void search\_element(ListE\* start, int element)

{

ListE\* curr = start;

int pos = 0;

while (curr->data != element && curr->next != nullptr)

{

curr = curr->next;

pos++;

}

if (curr->data == element)

{

cout << "Элемент найден" << endl;

}

else

{

cout << "Элемент не найден" << endl;

}

}

void printList(ListE\* start, int n)

{

ListE\* curr = start;

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

{

cout << curr->data << ' ';

curr = curr->next;

}

cout << endl;

}

void deleteList(ListE\* list)

{

delete list;

}

int main()

{

setlocale(LC\_ALL, "RUS");

int size, pos, k, element;

cout << "Введите размер списка: ";

cin >> size;

cout << "Введите значения элементов в списке: ";

ListE\* list = createList(size);

cout << "Вывод списка" << endl;

printList(list, size);

cout << "Введите на какую позицию хотите добавить элемент: ";

cin >> pos;

insert\_element(list, pos, 1, size);

cout << "Вывод списка" << endl;

printList(list, size);

cout << "Введите на какую позицию хотите добавить элементы и какое количество: ";

cin >> pos;

cin >> k;

insert\_elements(list, pos, k, size);

cout << "Вывод списка" << endl;

printList(list, size);

cout << "Введите на какой позиции хотите удалить элемент: ";

cin >> pos;

delete\_element(list, pos, 1, size);

cout << "Вывод списка" << endl;

printList(list, size);

cout << "Введите на какой позиции хотите удалить элементы и какое количество: ";

cin >> pos;

cin >> k;

delete\_elements(list, pos, k, size);

cout << "Вывод списка" << endl;

printList(list, size);

cout << "Какой элемент вы хотите найти?: ";

cin >> element;

search\_element(list, element);

deleteList(list);

return 0;

}

**Вывод программы**

