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 №11.2**

Дисциплина: основы алгоритмизации и программирования

Тема: Двунаправленные списки в С++

Вариант 24

Выполнил работу

студент группы РИС-20-1б

Коваленко Н. А.

Проверил

Доцент кафедры ИТАС

Полякова О.А.

Пермь, 2020

**Цель задачи**

Цель – написать программу для создания, редактирования и вывода двунаправленного списка. Реализовать цель через функции и стандартную stl библиотеку

Для достижения поставленной цели необходимо решить следующие задачи:

* Проанализировать логику двунаправленного списка
* Продумать, как будет создан двунаправленный список в с++
* Создать функцию, в котором будет создан первый элемент списка
* Создать функции для обработки однонаправленного списка
* Написать алгоритм, который будет редактировать исходный список

Постановка задачи

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

Анализ задачи

1. Определить какие операции должны быть выполнены по заданию:

* Создание структуры для описания узла двунаправленного списка и указателя на первый элемент

struct DoubleList

{

int data;

DoubleList \*next;

DoubleList \*prev;

};

DoubleList \*head;

* Описать функцию AddList, которая в качестве параметров принимает значение и адрес будущего узла, после чего создает его в списке:

void AddList(int value, int position)

* Написать основные функции для обработки двунаправленного списка
* Разобрать функции библиотеки stl для работы с двусвязными списками
* Реализовать алгоритм редактирования двусвязного списка через функции stl и созданные функции

1. Для решения задачи используются переменные:

Несколько целочисленных переменных для управления циклами и контроля размера списка

int numbersCount = -1;

int k, j=-1;

Указатели на следующие элементы списка

Node \*pNext;

1. Ввод данных осуществляется посредством функции

cin >> numbersCount;

Вывод данных осуществляется посредством функции

cout << "Измененный список: \t";

1. Для решения задачи будут использованы циклы. Например, цикл, который создает некоторое количество новых элементов в списке и заносит в них данные:

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

{

cin >> j;

newList.push\_back(j);

if (j % 2 == 0) k = i;

}

Для защиты программы от случайных ошибок были реализованы проверки входных данных:

Проверка размерности массива (размер больше 0):

int numbersCount = -1;

while (numbersCount <= 0)

{

cout << "Введи количество элементов в списке: ";

cin >> numbersCount;

}

1. Для описания основных операций редактирования списка используются функции, например функция для печати двусвязного списка:

void PrintList()

{

DoubleList \*a = head;

do

{

cout << a->data << " ";

a = a->next;

} while (a != head); cout << "\n\n";

## }

Код

#include <iostream>

#include <string>

#include <list>

using namespace std;

struct DoubleList

{

int data;

DoubleList \*next;

DoubleList \*prev;

};

DoubleList \*head;

void AddList(int value, int position)

{

DoubleList \*node = new DoubleList;

node->data = value;

if (head == NULL)

{

node->next = node;

node->prev = node;

head = node;

}

else

{

DoubleList \*p = head;

for (int i = position; i > 0; i--) p = p->next;

p->prev->next = node;

node->prev = p->prev;

node->next = p;

p->prev = node;

}

}

int DeleteList(int position)

{

if (head == NULL) { cout << "\nСписок пуст\n\n"; return 0; }

if (head == head->next)

{

delete head;

head = NULL;

}

else

{

DoubleList \*a = head;

for (int i = position; i > 1; i--) a = a->next;

if (a == head) head = a->next;

a->prev->next = a->next;

a->next->prev = a->prev;

delete a;

}

}

void PrintList()

{

DoubleList \*a = head;

do

{

cout << a->data << " ";

a = a->next;

} while (a != head); cout << "\n\n";

}

void double\_list()

{

int numbersCount = -1;

int k, j=-1;

while (numbersCount <= 0)

{

cout << "Введи количество элементов в списке: ";

cin >> numbersCount;

}

cout << "Введи элементы списка:" << endl;

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

{

cin >> k;

AddList(k, i);

if (k % 2 == 0) j = i+1;

}

if (j == -1)

cout << "Четных элементов нет";

else

{

DeleteList(j);

cout << "Измененный список: \t";

PrintList();

}

}

void double\_listSTL()

{

int numbersCount = -1, j, k;

while (numbersCount <= 0)

{

cout << "Введи количество элементов в списке: ";

cin >> numbersCount;

}

list<int> newList;

cout << "Введи элементы списка: " << endl;

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

{

cin >> j;

newList.push\_back(j);

if (j % 2 == 0) k = i;

}

auto it = newList.begin();

advance(it, k);

newList.erase(it);

cout << "Измененный список: " << endl;

for (auto i = newList.begin(); i != newList.end(); i++)

cout << \*i << " ";

}

int main()

{

setlocale(LC\_ALL, "ru");

int menu = -1;

while (menu != 1 && menu != 2 && menu != 3 && menu != 4 && menu != 5)

{

cout << "Выбери вид динамической структуры:" << endl << "\n" << "1) Двусвязный список" << "\n" << "2) Двусвязный список stl"

cout << "\nВвод: ";

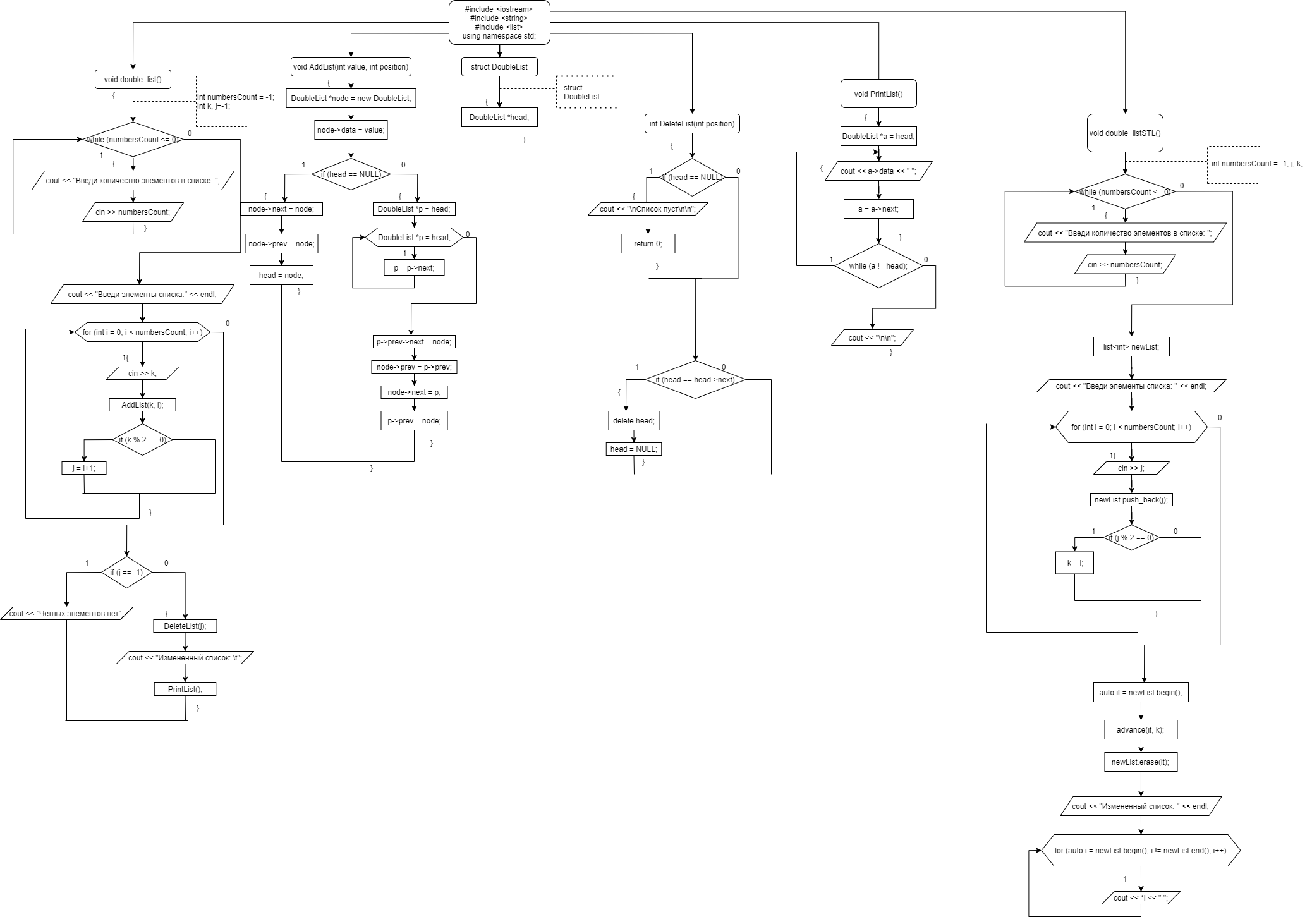
cin >> menu;

}

if (menu == 1) double\_list();

if (menu == 2) double\_listSTL();

Блок-схема:



Работа кода

