Министерство образования и науки Российской Федерации   
Федеральное государственное бюджетное образовательное учреждение   
высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра «Информационные технологии и автоматизированные системы

**Отчёт   
по лабораторной работе №11.1**

Дисциплина: основы алгоритмизации и программирования

Тема: «Двунаправленные списки»

Вариант №14

Выполнил работу:  
студент группы РИС-20-2б  
Вичугов Алексей Дмитриевич

Проверила:  
доцент кафедры ИТАС  
Полякова Ольга Андреевна

Пермь 2021

**Цели работы**

Получить практические навыки работы с двунаправленными списками;

**Постановка задачи**

1. Сформировать двунаправленный список, пользуясь STL «list». Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

1. Для решения задачи необходимо:
   1. Сформировать связанный двунаправленный:  
      list <int> List;

list <int> Nlist;

* 1. Распечатать созданный список:  
     cout << "Список: " << endl;

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

* 1. Удалить последний чётный элемент:  
      List.remove(u);

1. В ходе работы был использован типы данных int:  
    int d,i;
2. Данные для решения задачи представлены в виде переменных типа int:  
   int d,i=0;
3. Ввод данных в переменны производился с помощью оператора cin и функции push\_back():  
   cin >> d;

List.push\_back(d);

1. При решении поставленных задач использовались циклы:
   1. Для ввода данных и проверки правильности ввода цикл while:

while (d!=0){

cout << i++ << "-й элемент списка: " << endl;

cin >> d;

while (cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Должно быть введено число! Повторите ввод: " << endl;

cin >> d; }

**Полный код программы**

#include <iostream>

#include <list>

#include <iterator>

using namespace std;

list <int> List;

list <int> Nlist;

list <int>::iterator iter;

int u=0;

void Set(){

const int fon = system("Color F0");

system("chcp 1251>nul");

}

void Create(){

int d=1;

int i=1;

cout << "Введите элементы списка. Чтобы завершить ввод, введите 0. " << endl;

while (d!=0){

cout << i++ << "-й элемент списка: " << endl;

cin >> d;

while (cin.fail()){

cin.clear();

cin.ignore(30,'\n');

cout << "Должно быть введено число! Повторите ввод: " << endl;

cin >> d;

}

if (d!=0) List.push\_back(d);

if (d!=0&&d%2==0) u=d;

}

}

int main(){

Set();

Create();

int i=1;

cout << "Список: " << endl;

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

List.remove(u);

cout << endl << "Обновлённый список: " << endl;

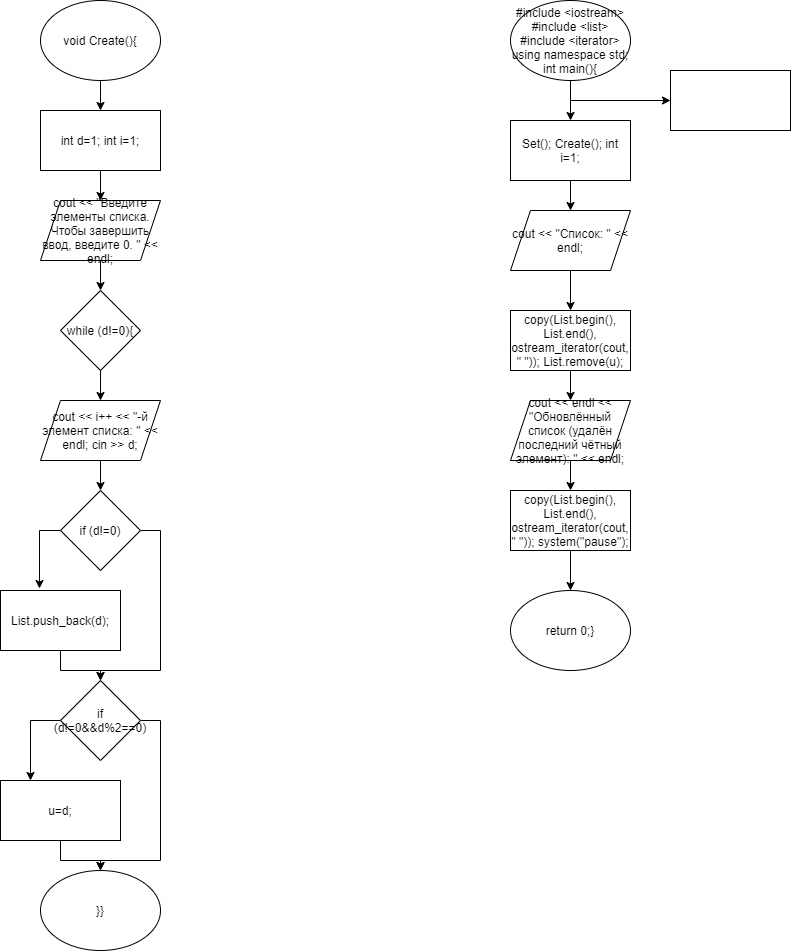
copy(List.begin(), List.end(), ostream\_iterator<int>(cout, " "));

system("pause");

return 0;

}

**Блок-схема**



**Результаты тестов**

