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. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

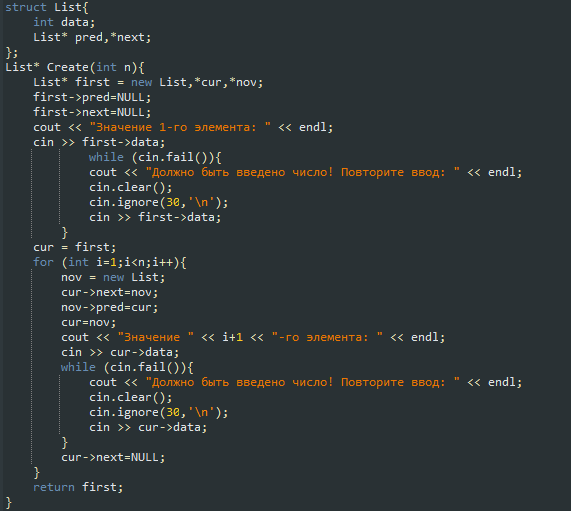
3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

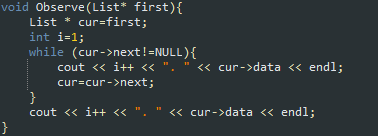
5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

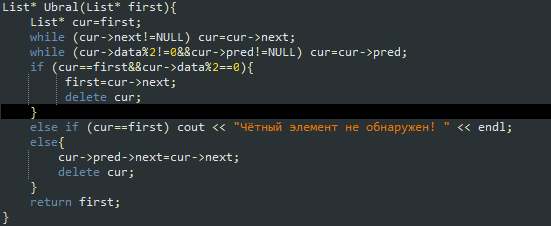
1. Для решения задачи необходимо:
   1. Сформировать связанный двунаправленный список из элементов структуры:



* 1. Сформировать функцию, распечатывающую данные созданного списка:



* 1. Сформировать функцию, реализующую удаление последнего чётного элемента в списке:



1. В ходе работы были использованы типы данных:
   1. int:  
       int data,i,n;
   2. Структура данных:  
      struct List{

int data;

List\* pred,\*next;

};

* 1. Указатели на тип данных List\*:  
      List \*cur,\*pred,\*next,\*first,\*nov,\*o,\*u;

1. Данные для решения задачи представлены в виде переменных типа int:  
   int n,i=0;
2. Структура, использованная в решении задачи содержит поля:

int data;

List\* pred,\*next;

1. Ввод данных в переменны производился с помощью оператора cin:  
   cin >> cur->data;  
   cin >> n;
2. При решении поставленных задач использовались циклы:
   1. Для проверки правильности ввода цикл while:

while (n<1){

if (cin.fail()){

cin.clear();

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

}}

* 1. Для формирования списка цикл for:

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

nov = new List;

cur->next=nov;

nov->pred=cur;

cur=nov;

cout << "Значение " << i+1 << "-го элемента: " << endl;

cin >> cur->data;

while (cin.fail()){

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

cin.clear();

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

cin >> cur->data;

}

cur->next=NULL;

}

**Полный код программы**

#include <iostream>

using namespace std;

struct List{

int data;

List\* pred,\*next;

};

List\* Create(int n){

List\* first = new List,\*cur,\*nov;

first->pred=NULL;

first->next=NULL;

cout << "Значение 1-го элемента: " << endl;

cin >> first->data;

while (cin.fail()){

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

cin.clear();

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

cin >> first->data;

}

cur = first;

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

nov = new List;

cur->next=nov;

nov->pred=cur;

cur=nov;

cout << "Значение " << i+1 << "-го элемента: " << endl;

cin >> cur->data;

while (cin.fail()){

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

cin.clear();

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

cin >> cur->data;

}

cur->next=NULL;

}

return first;

}

void Observe(List\* first){

List \* cur=first;

int i=1;

while (cur->next!=NULL){

cout << i++ << ". " << cur->data << endl;

cur=cur->next;

}

cout << i++ << ". " << cur->data << endl;

}

List\* Ubral(List\* first){

List\* cur=first;

while (cur->next!=NULL) cur=cur->next;

while (cur->data%2!=0&&cur->pred!=NULL) cur=cur->pred;

if (cur==first&&cur->data%2==0){

first=cur->next;

delete cur;

}

else if (cur==first) cout << "Чётный элемент не обнаружен! " << endl;

else{

cur->pred->next=cur->next;

delete cur;

}

return first;

}

void Set(){

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

system("chcp 1251>nul");

}

int main(){

Set();

List\* o,\*u;

int n;

cout << "Введите количество элементов списка: " << endl;

cin >> n;

while(n<2){

cout << "Неверный ввод! Список должен состоять минимум из 2-х элементов! Повторите ввод: " << endl;

cin.clear();

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

cin >> n;

}

o=Create(n);

Observe(o);

o=Ubral(o);

cout << "Обновлённый список (удалён последний чётный элемент): " << endl;

Observe(o);

while (o->next!=NULL){

u=o;

o=o->next;

delete u;

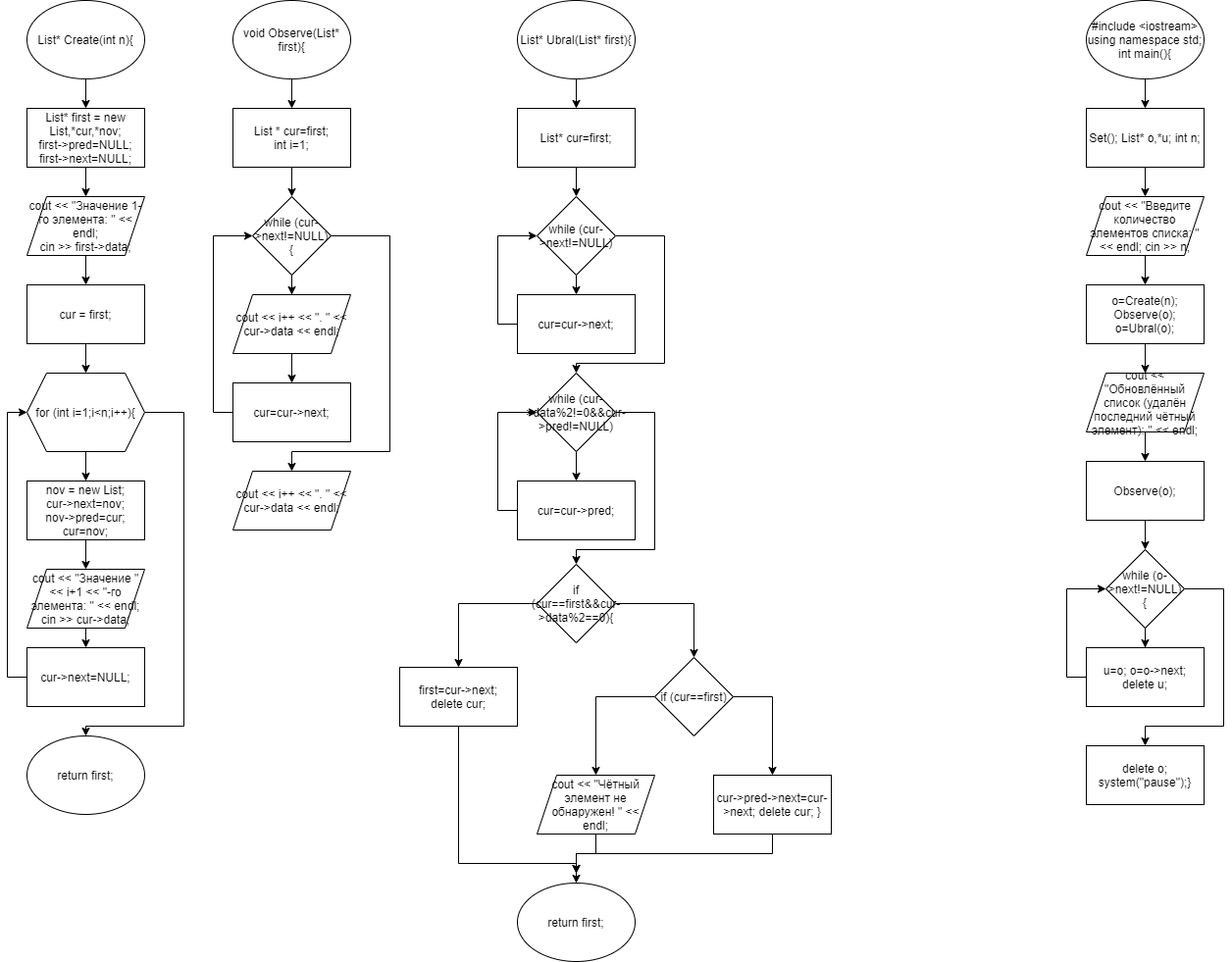
}

delete o;

system("pause");

}

**Блок-схема**



**Результаты тестов**

