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: «Однонаправленные 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 однонаправленными списками;

**Постановка задачи**

1. Сформировать однонаправленный список. Тип информационного поля указан в варианте.

2. Распечатать полученную структуру.

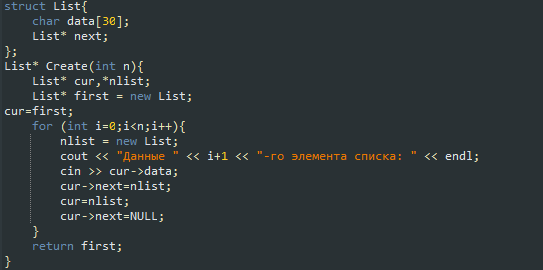
3. Выполнить обработку структуры в соответствии с заданием.

4. Распечатать полученный результат.

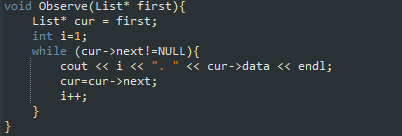
5. Удалить соответствующую структуру из памяти.

**Анализ задачи**

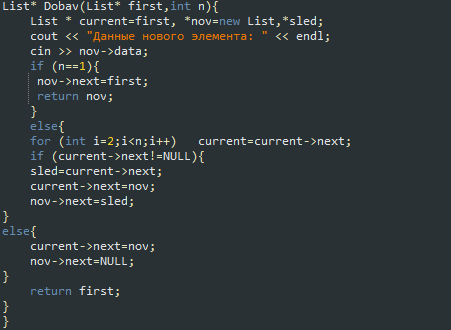
1. Для решения задачи необходимо:
   1. Сформировать связанный однонаправленный список из элементов структуры:



* 1. Сформировать функцию, распечатывающую данные созданного списка:



* 1. Сформировать функцию, реализующую добавление элемента в список по заданному номеру:



1. В ходе работы были использованs типы данных:
   1. int:  
       int n,d,i=0;
   2. Структура данных:  
      struct List{

char data[30];

List\* next;

};

* 1. Указатели на тип данных List\*:  
      List\* cur,nlist,newt,first,current;

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

char data[30];

List\* 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=0;i<n;i++){

nlist = new List;

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

cin >> cur->data;

cur->next=nlist;

cur=nlist;

cur->next=NULL;

}

**Полный код программы**

#include <iostream>

using namespace std;

struct List{

char data[30];

List\* next;

};

List\* Create(int n){

List\* cur,\*nlist;

List\* first = new List;

cur=first;

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

nlist = new List;

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

cin >> cur->data;

cur->next=nlist;

cur=nlist;

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;

i++;

}

}

void Set(){

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

system("chcp 1251>nul");

}

List\* Dobav(List\* first,int n){

List \* current=first, \*nov=new List,\*sled;

cout << "Данные нового элемента: " << endl;

cin >> nov->data;

if (n==1){

nov->next=first;

return nov;

}

else{

for (int i=2;i<n;i++) current=current->next;

if (current->next!=NULL){

sled=current->next;

current->next=nov;

nov->next=sled;

}

else{

current->next=nov;

nov->next=NULL;

}

return first;

}

}

int main(){

Set();

int n,d;

List\* o;

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

cin >> n;

while (n<1){

if (cin.fail()){

cin.clear();

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

}

cout << "Неверный ввод! Список должен иметь минимум 1 элемент! Повторите ввод. " << endl;

cin >> n;

}

o=Create(n);

Observe(o);

cout << "Введите порядковый номер добавляемого элемента. " << endl;

cin >> d;

while (d<1||d>n+1){

if (cin.fail()){

cin.clear();

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

}

cout << "Номер элемента должен быть больше 0 и не должен превышать количество элементов в списке более чем на 1. Повторите ввод. " << endl;

cin >> d;

}

o=Dobav(o, d);

Observe(o);

while (o->next!=NULL){

u=o;

o=o->next;

delete u;

}

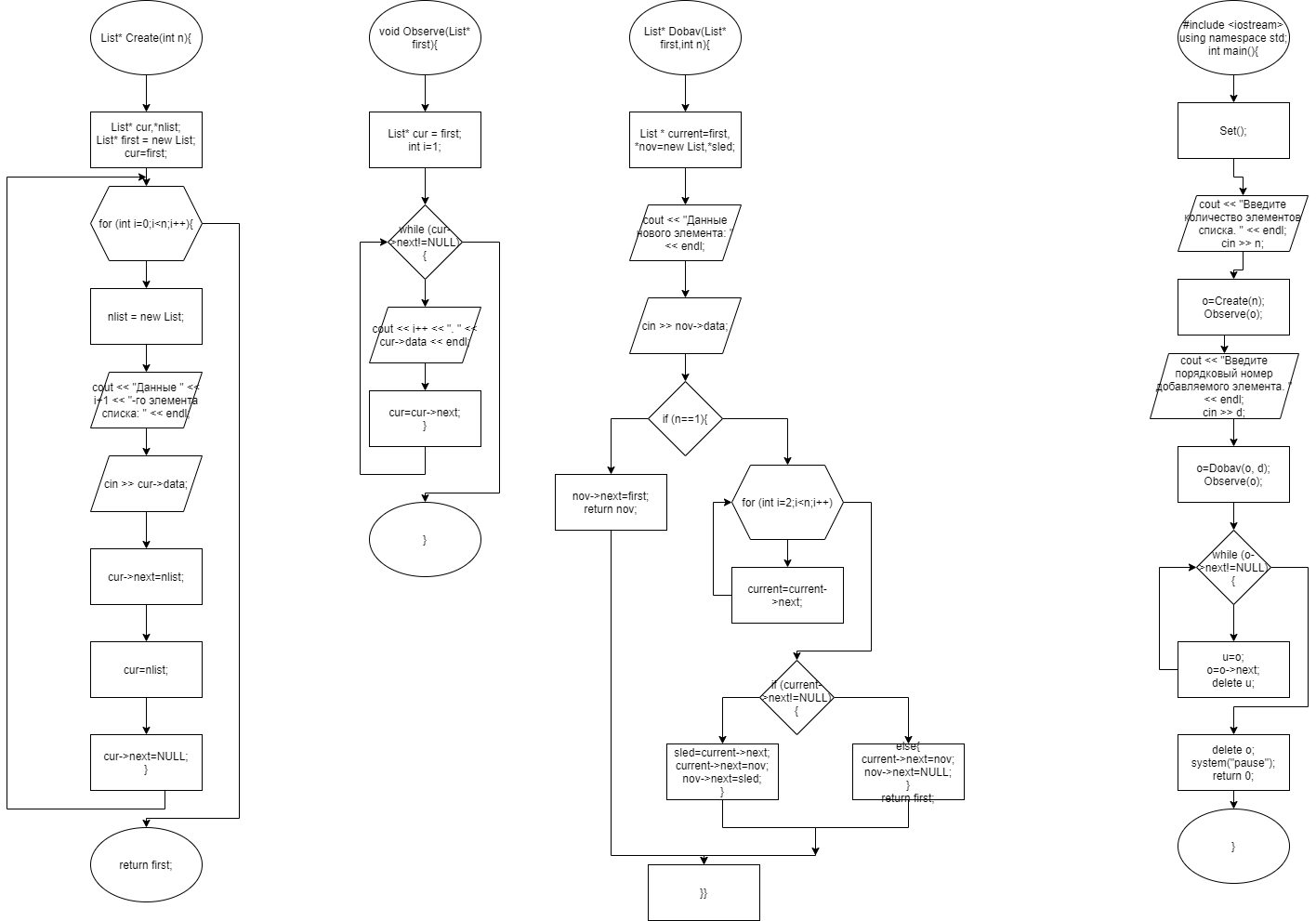
delete o;

system("pause");

return 0;

}

**Блок-схема**



**Результаты тестов**

