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

**Пермский национальный исследовательский политехнический университет**

Факультет Электротехнический Кафедра ИТАС

**ОТЧЁТ**

**о лабораторной работе № 11**

|  | Выполнил:  Студент группы ПРТ-21-1Б  Швалев Д.И.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Проверил:  Доцент кафедры ИТАС Полякова О. А.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| --- | --- |

**Пермь 2022**

## Цель лабораторной работы: применение динамических информационных структур, с использованием одно- и двунаправленых списков.

Постановка задачи: написать программу, в которой информационные создаются динамические структуры с одно- и двунаправленными списками и выполнить их обработку в соответствии со своим вариантом.

## 

## Текст программы:

#include <iostream>

#include <fstream>

using namespace std;

struct point

{

int key;

point\* next;

};

point\* make\_point(int n)

{

point\* first, \* p;

first = NULL;

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

{

p = new point();

cout << "Введите значение: ";

cin >> p->key;

p->next = first;

first = p;

}

return first;

}

void print(point\* first)

{

if (first == NULL)

cout << "Список пустой" << endl;

else

{

point\* p = first;

while (p != NULL)

{

cout << p->key << " ";

p = p->next;

}

cout << endl;

}

}

point\* get\_last(point\* first)

{

while (first->next != NULL)

{

first = first->next;

}

return first;

}

point\* delete\_element(point\* first, int number)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return first;

}

if (number < 1)

{

cout << "Неверный номер элемента" << endl;

return first;

}

if (number == 1)

{

point\* temp = first;

first = first->next;

delete temp;

}

else

{

int i = 1;

point\* p = first;

while (p != NULL && i < number - 1)

{

i++;

p = p->next;

}

if (p == NULL)

cout << "Неверный номер элемента" << endl;

else

{

point\* temp = p->next;

p->next = p->next->next;

delete temp;

}

}

return first;

}

point\* add\_elements(point\* first, int number, int k)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return first;

}

if (number < 1)

{

cout << "Неверный номер элемента" << endl;

return first;

}

if (number == 1)

{

point\* beg\_new\_list = make\_point(k);

point\* end\_new\_list = get\_last(beg\_new\_list);

end\_new\_list->next = first;

first = beg\_new\_list;

}

else

{

int i = 1;

point\* p = first;

while (p != NULL && i < number - 1)

{

i++;

p = p->next;

}

if (p == NULL)

cout << "Неверный номер элемента" << endl;

else

{

point\* beg\_new\_list = make\_point(k);

point\* end\_new\_list = get\_last(beg\_new\_list);

end\_new\_list->next = p->next;

p->next = beg\_new\_list;

}

}

return first;

}

void free\_list(point\*\* first)

{

point\* p = \*first, \* t;

while (p != NULL)

{

t = p->next;

delete p;

p = t;

}

\*first = NULL;

}

void save\_to\_file(point\* first, const char\* filename)

{

if (first == NULL)

{

cout << "Список пустой" << endl;

return;

}

ofstream file(filename);

point\* p = first;

while (p != NULL)

{

file << p->key;

if (p->next != NULL)

file << " ";

p = p->next;

}

file.close();

}

point\* read\_file(const char\* filename)

{

ifstream file(filename);

if (!file.is\_open())

{

cout << "Не удалось открыть файл" << endl;

return NULL;

}

point\* beg = NULL, \* p, \* pred = NULL;

int num;

while (!file.eof())

{

file >> num;

p = new point();

p->key = num;

p->next = NULL;

if (beg == NULL)

beg = p;

else

pred->next = p;

pred = p;

}

file.close();

return beg;

}

int main()

{

setlocale(LC\_ALL, "rus");

point\* beg = NULL;

int oper;

do

{

system("cls");

cout << "1.Создать список" << endl;

cout << "2.Удалить элемент с заданным номером" << endl;

cout << "3.Вставка элементов" << endl;

cout << "4.Вывод списка" << endl;

cout << "5.Сохранение в файл" << endl;

cout << "6.Загрузка из файла" << endl;

cout << "0.Выход" << endl;

cin >> oper;

switch (oper)

{

case 1:

{

int n;

cout << "Кол-во эл-тов списка = ";

cin >> n;

beg = make\_point(n);

print(beg);

break;

}

case 2:

{

print(beg);

int number;

cout << "Введите номер: ";

cin >> number;

beg = delete\_element(beg, number);

print(beg);

break;

}

case 3:

{

print(beg);

int number;

cout << "Введите номер: ";

cin >> number;

int k;

cout << "Сколько элементов добавить: ";

cin >> k;

beg = add\_elements(beg, number, k);

print(beg);

break;

}

case 4:

{

print(beg);

break;

}

case 5:

{

char filename[200];

cout << "Введите название файла: ";

cin >> filename;

save\_to\_file(beg, filename);

free\_list(&beg);

print(beg);

break;

}

case 6:

{

char filename[200];

cout << "Введите название файла: ";

cin >> filename;

beg = read\_file(filename);

print(beg);

break;

}

default:

break;

}

if (oper != 0)

system("pause");

} while (oper != 0);

free\_list(&beg);

}

## Ответ для варианта №4

