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**

Дисциплина: основы алгоритмизации и программирования

Тема: Информационные динамические структуры

Выполнил работу

студент группы МИР-21-2б

Поздняков Е.С.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Проверила

Доцент кафедры ИТАС

к.т.н. Полякова О. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Пермь 2022

**Цель работы**

Знакомство с динамическими информационными структурами на примере одно- и двунаправленных списков.

**Постановка задачи**

Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

Для каждого вариант разработать следующие функции:

1. Создание списка.
2. Добавление элемента в список (в соответствии со своим вариантом).
3. Удаление элемента из списка (в соответствии со своим вариантом).
4. Печать списка.
5. Запись списка в файл.
6. Уничтожение списка.
7. Восстановление списка из файла.

**Анализ задачи**

1. Создать структуру списка
2. Создать необходимые методы структуры для удаления из списка, добавления в него и чтения из него элементов
3. Объявить необходимые переменные
4. Записать в список данные от пользователя
5. Показать, что записано
6. Удалить указанные пользователем данные
7. Показать весь список
8. Записать в файл список
9. Почистить список, сопроводив соответствующим сообщением
10. Перезаписать или восстановить список из файла
11. Показать восстановленный список

**Решение**

#include <iostream>

#include <fstream>

using namespace std;

struct Node {

int val;

Node\* next;

Node(int \_val) : val(\_val), next(nullptr) {}

};

struct list {

Node\* first;

Node\* last;

list() : first(nullptr), last(nullptr) {}

bool is\_empty() {

return first == nullptr;

}

void push\_back(int \_val) {

Node\* p = new Node(\_val);

if (is\_empty()) {

first = p;

last = p;

return;

}

last->next = p;

last = p;

}

void print() {

if (is\_empty()) return;

Node\* p = first;

while (p) {

cout << p->val << " ";

p = p->next;

}

cout << endl;

}

Node\* operator[] (const int index) {

if (is\_empty()) return nullptr;

Node\* p = first;

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

p = p->next;

if (!p) return nullptr;

}

return p;

}

void remove\_first() {

if (is\_empty()) return;

Node\* p = first;

first = p->next;

delete p;

}

void remove\_last() {

if (is\_empty()) return;

if (first == last) {

remove\_first();

return;

}

Node\* p = first;

while (p->next != last) p = p->next;

p->next = nullptr;

delete last;

last = p;

}

void remove(int \_val) {

if (is\_empty()) return;

if (first->val == \_val) {

remove\_first();

return;

}

else if (last->val == \_val) {

remove\_last();

return;

}

Node\* slow = first;

Node\* fast = first->next;

while (fast && fast->val != \_val) {

fast = fast->next;

slow = slow->next;

}

if (!fast) {

cout << "This element does not exist" << endl;

return;

}

slow->next = fast->next;

delete fast;

}

};

int main() {

list l;

int ndel;

int koldel;

cout << "vvedite kollichestvo elementov spiska" <<endl;

int qq;

cin >> qq;

cout << "vvedite posledovatelno elementi spiska" <<endl;

int err;

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

cin >> err;

l.push\_back(err);

}

cout <<"vash spisok: "<< endl;

l.print();

cout << "vvedite nomer elementa s kotorogo udalit elementi" <<endl;

cin >> ndel;

cout << "vvedite kollichestvo elementov dlya udaleniua" <<endl;

cin >> koldel;

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

l.remove(ndel+i);

};

cout <<"vash spisok: "<< endl;

l.print();

int mass [qq-koldel];

for (int i=0; i<(qq-koldel); i++){

mass[i]=(l[i]->val);

}

// записываем данные из массива в файл

fstream fs;

string path = "mfile.txt";

fs.open(path,fstream::in | fstream:: out);

if (!fs.is\_open()){

cout << "error to open file" << endl;

}

else{

cout << "file //mfile.txt// open" << endl;

cout << "idet zapis v file" << endl;

for (int i=0; i<(qq-koldel); i++){

fs << mass[i] << " ";

fs.flush();

}

}

fs.close();

fs.open(path,fstream::in | fstream:: out);

if (!fs.is\_open()){

cout << "error to open file" << endl;

}

else{

cout << "file //mfile.txt// open" << endl;

cout << "chtenie is fila" << endl;

cout << endl << "Vostanovlennui spisok: " << endl;

string str;

while (!fs.eof()){

str = "";

getline (fs, str);

cout << str << endl;

}

}

fs.close();

return 0;

}

**Скриншоты работы программы**

