МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | М.А.Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2 |
| «ЛИНЕЙНЫЕ И ЦИКЛИЧЕСКИЕ СПИСКИ» |
| по курсу: Структуры и алгоритмы обработки данных |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2022

1. ***Цель работы***

Целью работы является изучение структур данных «линейный список» и «циклический список», а также получение практических навыков их реализации.

1. ***Задание***

Реализовать структуры данных «линейный список» и «циклический список» в соответствии с заданным вариантом. Дополнительно программа должна осуществлять следующие операции: 1) Добавление/удаление элемента в список (с клавиатуры); 2) Вывод исходного и результирующего списков на экран; 3) Если списки являются многочленами, в выводе должна быть отражена степень каждого элемента. Варианты задания приведены в таблице 2. Элементы последовательности, или коэффициенты многочлена (в зависимости от варианта) – числовые значения элемента списка, количество таких элементов списка равно длине последовательности, или количеству коэффициентов многочлена.

1. ***Вариант задания***

Text

Description automatically generated

1. ***Листинг программы, реализующей алгоритм:***

#include <iostream>

using namespace std;

struct Node {

int a, n;

Node\* next;

Node(int \_a, int \_n = 0) : a(\_a), n(\_n), next(nullptr) {}

};

struct list {

Node\* first;

Node\* last;

int length;

list(): first(nullptr), last(nullptr), length(0) {}

bool is\_empty() {

return first == nullptr;

}

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 \_a) {

if (is\_empty()) return;

if (first -> a == \_a) {

remove\_first();

length -= 1;

return;

}

else if (last -> a == \_a) {

remove\_last();

length -= 1;

return;

}

Node\* slow = first;

Node\* fast = first->next;

while (fast && fast -> a != \_a) {

fast = fast -> next;

slow = slow -> next;

}

if (!fast) {

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

return;

}

slow -> next = fast -> next;

delete fast;

length -= 1;

}

void add\_value(int value) {

if (value == 0) {

return;

}

length += 1;

push\_back(value);

}

void push\_back(int \_a) {

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

if (is\_empty()) {

first = p;

last = p;

return;

}

last -> next = p;

last = p;

}

void add\_power\_to\_nodes(){

Node\* p = first;

int current\_power = length - 1;

while(p){

p -> n = current\_power;

current\_power -= 1;

if(p == last){

break;

}

p = p -> next;

}

}

void print() {

if (is\_empty()) return;

Node\* p = first;

cout << endl;

cout << "Текущий список:" << endl;

while (p) {

if (p -> n == 0) {

if (p -> a > 0 && p != first) cout << "+ " << p -> a << " ";

else cout << p -> a << " ";

}

else if(p -> n == 1){

if (p -> a > 0 && p != first) cout << "+ " << p -> a << "x ";

else cout << p -> a << "x ";

}

else{

if (p -> a > 0 && p != first) cout << "+ " << p -> a << "x^" << p -> n << " ";

else cout << p -> a << "x^" << p -> n << " ";

};

if (p == last) break;

p = p -> next;

}

cout << endl;

}

void do\_lab\_task(list first, list second){

cout << "Многочлен №1: ";

first.print();

cout << endl;

cout << "Многочлен №2: ";

second.print();

cout << endl;

if (first.length != second.length){

cout << "Многочлены не равны, так как не равны их длины" << endl;

}

else {

if (Equal(first, second)) {

cout << "Многочлены равны" << endl;

}

else {

cout << "Многочлены не равны" << endl;

}

}

}

void open\_menu(){

cout << "Меню:" << endl;

cout << "1 - Добавить элемент в список" << endl;

cout << "2 - Удалить элемент из списка" << endl;

cout << "3 - Перейти к другому многочлену" << endl;

cout << endl;

}

void do\_task\_from\_menu(int chosen\_task){

int value;

switch (chosen\_task) {

case 1:

cout << "Вы выбрали добавить элемент в список" << endl;

cout << "Введите значение многочлена: ";

cin >> value;

add\_value(value);

break;

case 2:

cout << "Вы выбрали удаление элемента" << endl;

cout << endl;

cout << "Выберите коэффицент, который вы хотите удалить: ";

int num\_node;

cin >> num\_node;

remove(num\_node);

break;

default:

break;

}

}

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;

}

bool Equal(list first, list second){

for (int i = 0; i < first.length; i++){

if (!(first[i] -> a == second[i] -> a) || !(first[i] -> n == second[i] -> n)) return false;

}

return true;

}

};

int main(){

list first, second;

int chosen\_task;

bool menu = true;

first.open\_menu();

while(menu){

cout << "Выберите действие: ";

cin >> chosen\_task;

first.do\_task\_from\_menu(chosen\_task);

if (chosen\_task == 3){

menu = false;

first.add\_power\_to\_nodes();

}

}

chosen\_task = 0;

menu = true;

cout << "Вы перешли к работе со вторым многочленом" << endl;

second.open\_menu();

while(menu){

cout << endl;

cout << "Выберите действие: ";

cin >> chosen\_task;

second.do\_task\_from\_menu(chosen\_task);

if (chosen\_task == 3){

menu = false;

second.add\_power\_to\_nodes();

}

}

cout << "Выполняем задание лабораторной работы" << endl;

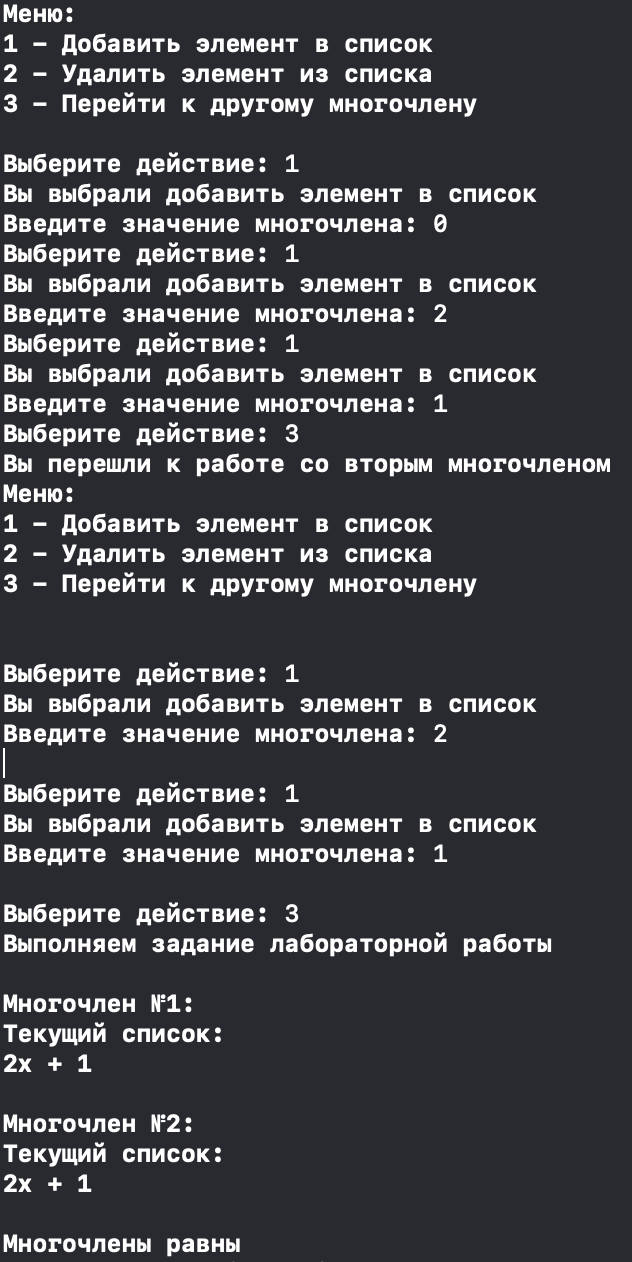
cout << endl;

second.do\_lab\_task(first, second);

return 0;

}

1. ***Примеры***

******

1. ***Вывод***

В ходе лабораторной работы я изучил структуры данных «линейный список» и «циклический список», а также получид практические навыки их реализации.