МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. И. С. ТУРГЕНЕВА»

Кафедра «Программная инженерия»

**ОТЧЁТ**

по лабораторной работе № 5

по дисциплине: «Объектно-ориентированное программирование

на С++»

Выполнил: Шорин В. Д. Шифр: 171406

Институт приборостроения, автоматизации и информационных технологий

Направление: 09.03.04 «Программная инженерия»

Группа: 71-ПГ

Проверил:

Отметка о зачете:

Дата: «\_\_» \_\_\_\_\_\_\_\_ 2018г.

Орёл. 2018

**Вариант 25**

**Задание**

Разработать класс для работы с многочленами. Реализовать набор операций для работы с классом (перегруженные операции):1)сложение,2)комбинированное вычитание 3)декремент постфиксный 4)и декремент префиксный (уменьшение значений всех коэффициентов на 1)5)преобразование к разным типам (от основного к пользовательскому, от пользовательского к основному).6)Перегрузить операции ввода/вывода, позволяющие осуществлять ввод и вывод объектов класса в удобной форме: ввод с клавиатуры и вывод на экран;7)в файл и из файла.8) В файл сохранять все созданные объекты. При загрузке программы их считывать.

**Код**

#include <iostream>#include <vector>#include <windows.h>#include <fstream>using namespace std;

class Polinom{

private: int n; //Старшая степень vector <double> coefs; //Коэффициенты

public:

Polinom(int N, int a){ n = N; for (int i = 0; i < N + 1; i++) { coefs.push\_back(0); } }

Polinom(int N){ n = N; double a; for(int i = 0; i < n + 1; i++) { cout << "Введите коэффициент при " << n - i << " степени: "; cin >> a;

coefs.push\_back(a); } }

void ShowInfo(){ for (int i = 0; i < coefs.size(); i++) { if(coefs.size() - i - 1 == 0){cout << coefs[i];continue;} if(coefs[i] == 0){continue;} if(coefs[i] == 1){cout << "x^" << coefs.size() - i - 1 << " + ";continue;} cout << coefs[i] << "x^" << coefs.size() - i - 1 << " + "; } }

void setCoefs(){double a; for(int i = 0; i < n + 1; i++) { cout << "Введите коэффициент при " << n - i << " степени: "; cin >> a; coefs.push\_back(a); } } void ShowN(){cout << n;}

void ShowCoefs(){ for(int i = 0; i < coefs.size(); i++) { cout << i << " coef is " << coefs[i] << endl; } }

double getCoef(int i){return coefs[i];} int getN(){return n;} friend Polinom operator+(Polinom , Polinom );

Polinom operator-=(const int& a); Polinom& operator--(); Polinom operator--(int);

friend ostream& operator<<(ostream& os, const Polinom& p); friend istream& operator>>(istream& is, Polinom& p);

friend ofstream& operator<<(ofstream& ofs, const Polinom& p);

**///Перегрузка файлового ввода (из файла)**

friend ifstream& operator>>(ifstream& ifs, Polinom& p){ for (int i = 0; i < p.n + 1; i++) { ifs >> p.coefs[i]; } return ifs; }

**operator int**(){ int res = 1; for(int i = 0;i < n + 1; i++) { res \*= (int)coefs[i]; } return res; }

**operator float**(){ float res = 1; for(int i = 0;i < n + 1; i++) { res \*= (float)coefs[i]; } return res; } ~Polinom(){coefs.clear();}};

**///Перегрузка сложения**

Polinom operator +(Polinom polim1, Polinom polim2){ if (polim1.getN() > polim2.getN()) { for(int i = 0; i < polim1.getN() + 1; i++)

{ polim1.coefs[polim1.getN() - polim2.getN() + i] = polim1.coefs[polim1.getN() - polim2.getN() + i] + polim2.coefs[i]; } return polim1; }

else if (polim1.getN() < polim2.getN()) { for(int i = 0; i < polim1.getN() + 1; i++)

{ polim2.coefs[polim2.getN() - polim1.getN() + i] = polim1.coefs[i] + polim2.coefs[polim2.getN() - polim1.getN() + i]; } return polim2; }

else { for(int i = 0; i < polim1.getN() + 1; i++) { polim1.coefs[i] = polim1.coefs[i] + polim2.coefs[i]; } return polim1; }}

**///Перегрузка комбинированного вычитания**Polinom Polinom::operator -=(const int& a){ for(int i = 0; i < n + 1; i++) { coefs[i] -= a; } return \*this;}

**///Перегрузка префиксного декремента**

Polinom& Polinom::operator --(){ for(int i = 0; i < this->getN() + 1; i++) { this->coefs[i] -= 1; } return \*this;}

**///Перегрузка постфиксного декремента**Polinom Polinom::operator --(int){ Polinom cur = \*this; for(int i = 0; i < this->getN() + 1; i++)

{ this->coefs[i] -= 1; } return cur;}

**///Перегрузка консольного вывода**

ostream& operator<<(ostream& os, const Polinom& p) \

for (int i = 0; i < p.coefs.size(); i++) { if(p.coefs.size() - i - 1 == 0){os << p.coefs[i];continue;} if(p.coefs[i] == 0){continue;}

if(p.coefs[i] == 1){os << "x^" << p.coefs.size() - i - 1 << " + ";continue;} os << p.coefs[i] << "x^" << p.coefs.size() - i - 1 << " + "; }

os << endl; return os;

**///Перегрузка консольного ввода**

istream& operator>>(istream& is, Polinom& p){

double a; for(int i = 0; i < p.n + 1; i++) { cout << "Введите коэффициент при " << p.n - i << " степени: "; is >> a; p.coefs.push\_back(a); } return is;}

**///Перегрузка файлового вывода (в файл)**

ofstream& operator<<(ofstream& ofs, const Polinom& p){ ofs << p.n << endl; for (int i = 0; i < p.n + 1; i++) { ofs << p.coefs[i] << " "; } ofs << endl; return ofs;}

**int main**(){

SetConsoleCP(1251); SetConsoleOutputCP(1251);

string way = "file.txt"; ifstream fin; fin.open(way, ios\_base::in);

if (!fin.is\_open()){cerr << "Error: file " << way << "is not open\n";}

vector <Polinom> curPolinoms;

int N = 1; int d; int i = 0; fin >> d;

for (i = 0; i < d; i++) { fin >> N; curPolinoms.push\_back(Polinom(N, N)); fin >> curPolinoms[i]; } int exit = 1; char usersChoise;

while(exit) { system("cls");

if(!curPolinoms.empty()) { cout << "Доступные многочлены: " << endl << endl;

for(int i = 0; i < curPolinoms.size(); i++) { cout << i + 1 << ")"; cout << curPolinoms[i]; cout << endl; } }

cout << endl << "Выберите действие: " << endl; cout << "Добавить новый многочлен - 1" << endl;

cout << "Сложить два многочлена - 2" << endl; cout << "Уменьшить на 1 коэффициенты определённого многочлена - 3" << endl;

cout << "Уменьшить на какое-то число коэффициенты определённого многочлена - 4" << endl; cout << "Привести к типу - 5" << endl << endl; cout << "Выход - 0" << endl << endl; cout << "Ваш выбор: "; cin >> usersChoise;

switch(usersChoise) {

case '1': { system("cls"); int n; cout << "Введите старшую степень многочлена: "; cin >> n;

curPolinoms.push\_back(Polinom(n)); cout << curPolinoms[curPolinoms.size() - 1]; system("pause"); break; }

case '2': { int id1, id2, exit1 = 1, exit2 = 1; cout << "Что складывать?" << endl;

while(exit1) {

cout << "Номер первого многочлена: "; cin >> id1;

if ( id1 - 1 < 0 || id1 - 1 > curPolinoms.size()){ cout << "Неправильно введён номер" << endl; continue; }

else {exit1 = 0;} }

while(exit2) {

cout << "Номер второго многочлена: "; cin >> id2;

if ( id2 - 1 < 0 || id2 - 1 > curPolinoms.size()) { cout << "Неправильно введён номер" << endl; continue; }

else {exit2 = 0;}

}

Polinom rezPolinom = curPolinoms.at(id1 - 1) + curPolinoms.at(id2 - 1); cout << endl << "Результат сложения: " << endl << rezPolinom;

system("pause"); break;

}

case '3': { int id, curExit = 1; while(curExit) {

cout << "Введите номер нужного многочлена: " << endl; cin >> id;

if(id - 1 < 0 || id - 1 > curPolinoms.size()) { cout << "Неправильно введён номер" << endl; continue; }

else {curExit = 0;} }

--curPolinoms.at(id - 1); cout << curPolinoms.at(id - 1); system("pause"); break; }

case '4': { int id, curExit = 1; double a;

while(curExit) { cout << "Введите номер нужного многочлена: " << endl; cin >> id;

if(id - 1 < 0 || id - 1 > curPolinoms.size()) { cout << "Неправильно введён номер" << endl; continue; }

else {curExit = 0;}

}

cout << "Введите число, на которое нужно уменьшить этот многочлен: "; cin >> a; curPolinoms.at(id - 1) -= a;

cout << "Результат: " ; cout << curPolinoms.at(id - 1); system("pause"); break;

}

case '5': {

char us; cout << endl << "INT - 1" << endl; cout << "FLOAT - 2" << endl; cin >> us;

switch(us){

case '1': { int a; int proizv = 1; cout << "Введите номер многочлена: " ; cin >> a;

proizv = static\_cast<int>(curPolinoms[a - 1]); cout << "P = " << proizv << endl; break; }

case '2': { int a; float proizv = 1; cout << "Введите номер многочлена: " ; cin >> a; proizv = static\_cast<float>(curPolinoms[a - 1]); cout << "Proizvedenie = " << proizv << endl; break; } } system("pause"); break; }

case '0': { exit = 0; break;} } }

ofstream ft(way, ios\_base::out);

ft.close();

ofstream fout(way, ios\_base::app);//, ios\_base::app - для записи в конец файла

if (!fout.is\_open()){cerr << "Error: file " << way << "is not open\n";}

fout << curPolinoms.size() << endl;

for(int i = 0; i < curPolinoms.size(); i++) { fout << curPolinoms[i]; }

fout.close(); fin.close(); return 0;}