Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики

Базовая кафедра вычислительных и информационных технологий

**ОТЧЕТ О ТЕХНОЛОГИЧЕСКОЙ  
(ПРОЕКТНО-ТЕХНОЛОГИЧЕСКОЙ) ПРАКТИКЕ**

**Разработка консольного приложения «Polymaniak» средствами языка С++**

Руководитель \_\_\_\_\_\_\_\_\_ И. В. Баженова

подпись, дата

Студент ИМ23-02Б \_\_\_\_\_\_\_\_\_ Т. С. Байдалиев

группа, № зачетной книжки подпись, дата

Студент ИМ23-02Б \_\_\_\_\_\_\_\_\_ С. П. Лахин

группа, № зачетной книжки подпись, дата

Студент ИМ23-02Б \_\_\_\_\_\_\_\_\_ М.П. Кормилец

группа, № зачетной книжки подпись, дата

Красноярск 2024

**Общие сведенья**

Полиномом (многочленом) называют **выражение** вида

,

где , где - произвольное кольцо с операциями +, \*.

Минимальное натуральное (или ноль) число *n* такое, что для любого *m > n : ,* называют *степенью многочлена.*

Многочлен, у которого для всех , имеет степень 0 по определению.

Над многочленами определены операции +, \*, взятия производной, нормирования.

**Класс polynomial**

#include <polynomiac\_basic.hpp>

Класс polynomial представляет собой полином составленный из действительных чисел.

Каждый метод реализован соответственно с ТЗ.

**Методы класса:**

polynomial (void); - конструктор по умолчанию

polynomial (std::istream&); - считывания полинома со строки csv файла, потока std::istream

polynomial (const polynomial &);

- конструктор копирования

polynomial (std::initializer\_list<double>);

- конструктор для явной инициализации

polynomial (int range\_min, int range\_max, unsigned deg);

- генерация многочлена степени deg с коэффицентами в диапозоне от range\_min до range\_max

polynomial (unsigned dg);

- инициализация с преаллокацией

polynomial (std::vector<double>);

- инициализация многочлена с коэффициентами в std::vector

polynomial **operator+** (const polynomial&) const;

- сложение многочленов

polynomial **operator-** (const polynomial&) const;

- вычитание многочленов

polynomial **operator+** () const;

- Унарный плюс

polynomial **operator-** () const;

- Отрицательный многочлен

polynomial **operator\*** (const polynomial&) const;

- произведение многочленов

polynomial **operator/** (const polynomial&) const;

- деление многочленов (основная часть)

polynomial **operator%** (const polynomial&) const;

- деление многочленов (остаток)

polynomial **operator+=**(const polynomial&);

- прибавление многочлена с присваиванием

polynomial **operator-=**(const polynomial&); - вычитание многочлена с присваиванием

polynomial **operator\*=**(const polynomial&); - умножение многочленов с присваиванием

polynomial **operator/=**(const polynomial&); - деление многочленов с присваиванием

polynomial **operator%=**(const polynomial&); - остаток от дедения многочленов с присваиванием

polynomial **derivative**(void) const ; - производная многочлена

polynomial& **operator=** (const polynomial&);

- присваивание многочлена

bool **operator==**(const polynomial&) const;

- сравнение многочленов

bool **operator!=**(const polynomial&) const;

- сравнение многочленов

double **at** (unsigned) const;

- возвращает значение соответствующего коэффициента (аналог геттера)

void **set** (std::initializer\_list<double>);

void **set** (unsigned, double); - методы замены коэффициентов

double **operator()** (double x) const;

- значение полинома в данной точке x

friend unsigned **deg** (const polynomial&);

- функция, возращающая степень данного ей многочлена

void **special\_out** (std::ostream& out) const;

- специальный вывод для csv файла

~polynomial (void);

- деструктор

Дополнительные объявления:

const polynomial **ZERO** = polynomial{0};

- нуль-многочлен (многочлен степени 0, все коэффициэнты равны 0)

polynomial **x\_monomial** (const unsigned, const double);

- функция, возвращающая **одночлен** определённой степени и с определённым коэффицентом

**Класс PolynomialOperations**

#include <PolynomialOperations.hpp>

Класс PolinomialOperations содержит в себе методы, позволяющие совершать операции над многочленами:

static polynomial **sum** (const polynomial& a, const polynomial& b);

- возвращает сумму полиномов a и b

static polynomial **minus** (const polynomial& a);

- возвращает полином a с отрицательными коэффициэнтами

static polynomial **dif** (const polynomial& a, const polynomial& b);

- возвращает a + (-b)

static polynomial **mult** (const polynomial& a, const polynomial& b);

- возвращает произвдение данных полиномов

static polynomial **mod**(const polynomial& a, const polynomial& b);  
 - возвращает остаток от деления полинома a на b

static polynomial **div** (const polynomial& a, const polynomial& b);

- возвращает целую часть частного деления полинома a на b

static polynomial **derivative** (const polynomial& a);

- возвращает произволную данного полинома

static polynomial **normalize**(const polynomial& p);

- нормализирует данный полином

static bool **div\_ok**(const polynomial& a, const polynomial& b);

- возвращает истинное значение, если deg(b) < deg(a), ложное - в противном случае

static bool **degree\_equal**(const polynomial& a, const polynomial& b);

- возвращает истинное значение, если степени предоставленных полиномов равны, ложное — в противном случае

static void **divisor\_not\_null**(const polynomial &b);

- возвращает истинное значение, если b != ZERO, ложное - в противноим случае

Этот класс необходимо было создать по ТЗ. Операторы +, -, \*, /, % так же перегруженны для класса polynomial. Рекомендуется использовать именно их.

Имеются так же операторы с присваиванием.

|  |  |
| --- | --- |
| Запись через класс PolynomialOperations | Запись через операторы |
| PolynomialOperations.sum(a, b) | a+b |
| PolynomialOperations.dif(a, b) | a-b |
| PolynomialOperations.mod(a, b) | a\*b |
| PolynomialOperations.div(a, b) | a/b |
| PolynomialOperations.mod(a, b) | a%b |
| PolynomialOperations.minus(a) | -a |

Таблица эквивалентности методов и операторов

**Класс MenuManager**

Класс MenuManager предоставляет интерфейс для взаимодействия с пользователем, позволяя добавлять, удалять и выполнять операции над полиномами.

# Члены класса

Публичные методы

1. **void run();**

Описание: Запускает интерфейс управления. Этот метод отвечает за отображение меню и обработку выбора пользователя. Он будет запрашивать у пользователя различные действия и вызывать соответствующие методы для их выполнения.

Приватные методы

1. **void addRandomPolynomial(int range\_min, int range\_max, unsigned degree);**

- добавляет случайный полином в вектор.

Параметры:

*int range\_min* — Минимальное значение для коэффициентов полинома.

*int range\_max* — Максимальное значение для коэффициентов полинома.

*unsigned degree* — Степень полинома.

1. **void addPolynomialFromInput();**

Описание: Добавляет полином из входных данных. Этот метод запрашивает у пользователя ввод коэффициентов полинома и добавляет его в вектор.

1. **void removePolynomial();**

- удаляет полином из вектора. Этот метод запрашивает у пользователя номер полинома для удаления и удаляет соответствующий полином из вектора.

1. **void listPolynomials();**

- выводит список всех полиномов. Этот метод отображает все полиномы, хранящиеся в векторе, с их номерами и коэффициентами.

1. **void performOperations();**

- выполняет операции над выбранными полиномами. Этот метод запрашивает у пользователя номера двух полиномов и выполняет над ними операции: сложение, вычитание, умножение, деление и вычисление производной.

1. **void printPolynomials() const;**

- выводит все полиномы в векторе. Этот метод отвечает за форматированный вывод всех полиномов, хранящихся в векторе, на экран.

**Класс PolynomialOperations**

Класс предоставляет статические методы для выполнения операций над полиномами.

Методы

1. **static polynomial sum(const polynomial& a, const polynomial& b)**

Описание: Возвращает сумму двух полиномов.

Параметры:

**a**: Первый полином.

**b**: Второй полином.

Возвращает: Полином, представляющий сумму a и b.

1. **static polynomial minus(const polynomial& a)**

Описание: Возвращает унарное вычитание полинома (изменяет знак).

Параметры:

**a**: Полином.

Возвращает: Полином, представляющий -a.

1. **static polynomial dif(const polynomial& a, const polynomial& b)**

Описание: Возвращает разность двух полиномов.

Параметры:

**a**: Первый полином.

**b**: Второй полином.

Возвращает: Полином, представляющий a - b.

1. **static polynomial mult(const polynomial& a, const polynomial& b)**

Описание: Возвращает произведение двух полиномов.

Параметры:

**a**: Первый полином.

**b**: Второй полином.

Возвращает: Полином, представляющий a \* b.

1. **static polynomial mod(const polynomial& a, const polynomial& b)**

Описание: Вычисляет остаток от деления одного полинома на другой.

Параметры:

**a**: Делимое.

**b**: Делитель.

Возвращает: Остаток от деления a на b.

1. **static polynomial div(const polynomial& a, const polynomial& b)**

Описание: Возвращает частное от деления одного полинома на другой.

Параметры:

**a**: Делимое.

**b**: Делитель.

Возвращает: Частное от деления a на b.

1. **static polynomial derivative(const polynomial& a)**

Описание: Вычисляет производную полинома.

Параметры:

**a**: Полином.

Возвращает: Полином, представляющий производную a.

1. **static polynomial normalize(const polynomial& p)**

Описание: Нормализует полином, приводя его к стандартному виду.

Параметры:

**p**: Полином.

Возвращает: Нормализованный полином.

1. **static bool div\_ok(const polynomial& a, const polynomial& b)**

Описание: Проверяет, возможно ли деление полиномов.

Параметры:

**a**: Делимое.

**b**: Делитель.

Возвращает: *true*, если деление возможно, иначе *false*.

1. **static bool degree\_equal(const polynomial& a, const polynomial& b)**

Описание: Проверяет, имеют ли два полинома одинаковую степень.

Параметры:

**a**: Первый полином.

**b**: Второй полином.

Возвращает: *true*, если степени равны, иначе *false*.

1. **static bool divisor\_not\_null(const polynomial& b)**

Описание: Проверяет, является ли делитель ненулевым полиномом.

Параметры:

**b**: Делитель.

Возвращает: *true*, если b не нулевой, иначе *false*.

# Использование программы

При запуске программы вам будет предоставлен выбор команд.

1. Создать рандомный полином
2. Ввести новый полином
3. Ввести полином из файла

3. Удалить полином

4. Вывести список полиномов

5. Операции над полиномами

0. Выход

1. «**Создать рандомный полином**»

Команда для создания полинома по заданным параметрам минимального коэффициента, максимального коэффициента и максимальной степени переменных будет выводить полином с рандомными коэффициентами из заданных передов (min=<n<max)

1. «**Ввести новый полином**»

Команда для ввода полинома из консоли в ручную. Для использования нужно ввести коэффициенты полинома, от младшей степени к старшей. Далее уже готовый полином будет выведен на утверждение и запись или удаление.

1. «**Ввести полином из файла**»

Команда для дополнения списка полиномов из файла. Из консоли нужно задать имя файла полностью Пример: “имя\_файла.csv” Производится считывание **CSV** файла

1. «**Удалить полином**»

Команда для удаления полинома. Будет выведен список всех полиномов с номерами. При введении определённого номера данный полином будет удалён из списка без возможности восстановления.

1. «**Вывести список полиномов**»

Команда для выведения актуального списка доступных для работы полиномов.

1. «**Операции над полиномами**»

Команда для вывода меню с возможными действия ми над полиномами.

Пользователю будет предоставлен список полиномов и возможность выбрать 2 из этого списка с помощью их номеров.

Далее будут выведены результаты всех возможных операций над выбранными полиномами.

Производимые операции:

* Сложение
* Вычитание
* Умножение
* Деление (если возможно)
* Взятие производной первого полинома

1. «**Очистить экран**»

Чистит консоль от предыдущих операций.

# Тестирование

# photo_2024-12-26_16-37-40

Рис 1. «Начальное меню»

Началом тестирования стало создание полиномов по пунктам.

1. При рандомном создании сознательно была допущена ошибка. На Рис 2 Показано, что ошибка обрабатывается в виде исключения и программа продолжает работать штатно.

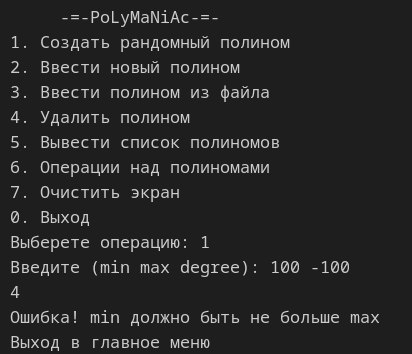


Рис 2. «Обработка ошибки»

При вводе корректных данных программа выдаёт нужный нам полином на утверждение и записывает его в список полиномов при согласии. См Рис 3

# photo_2024-12-26_16-37-33

Рис 3. «Корректные данные для рандомного заполнения»

1. Ввод полинома в ручную производится в в формате его коэффициентов при младших -> старших степенях. После заполнения коэффициентов происходит вывод созданного полинома и утверждение оного на запись в список. Рис 4 дано согласие на сохранение полинома Рис 5 создание полинома отменено

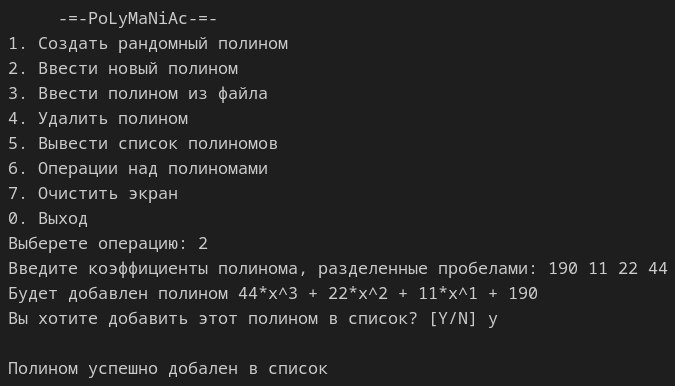


Рис 4. «Введение полинома с клавиатуры»

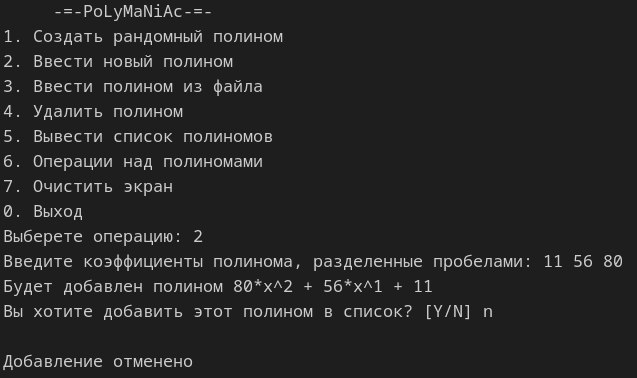


Рис 5. «Отказ от записи полинома»

1. Ещё одним вариантом задавать полиномы является считывание их из форматного txt файла. При некорректном вводе пути и имени файла обрабатывается ошибка. Рис 6

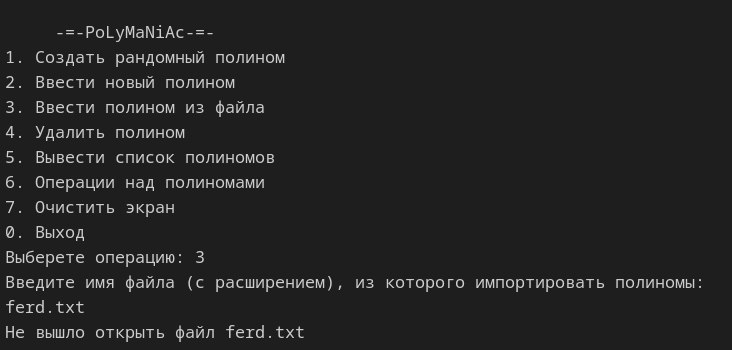


Рис 6. «Обработка ошибки считывания файла»

При вводе существующего форматного файла происходит создание полиномов по количеству файлов, а также их вывод на утверждение и запись. Рис 7

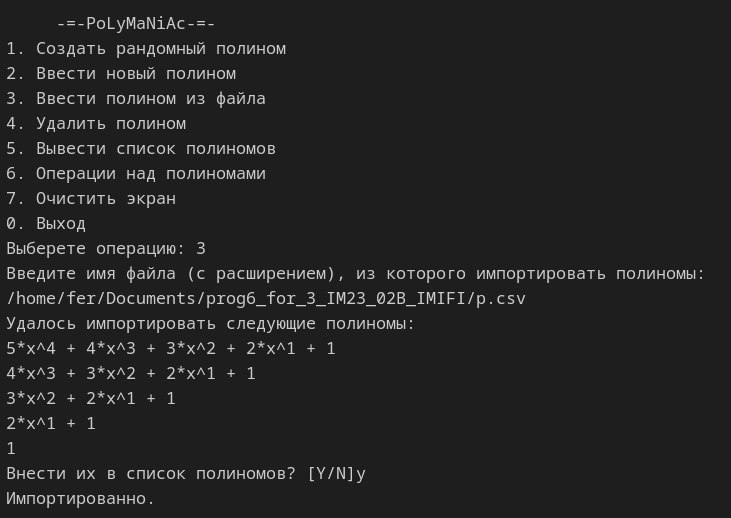


Рис 7. «Ввод полиномов из файла»

4) После создания некоторого количества полиномов можно протестировать возможность удаления полинома из списка, для этого выбираем четвёртый пункт меню и выбираем номер удаляемого полинома. Если выбрать номер выходящей за список полиномов это также будет обрабатываться ошибкой Рис 8

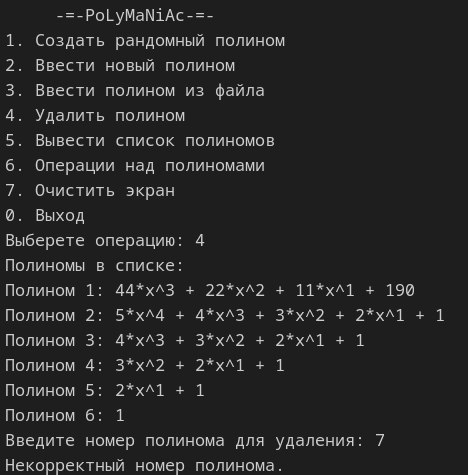
1. 

Рис 8. «ввод некорректного номера на удаление»

При введении корректного номера полином удаляется из списка полиномов безвозвратно. Рис 9

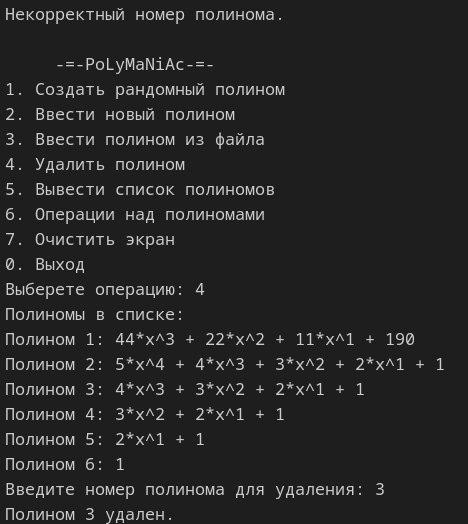


Рис 9. «Удаление полинома из списка»

1. Пятым пунктом выводится полный актуальный список полиномов Рис 10

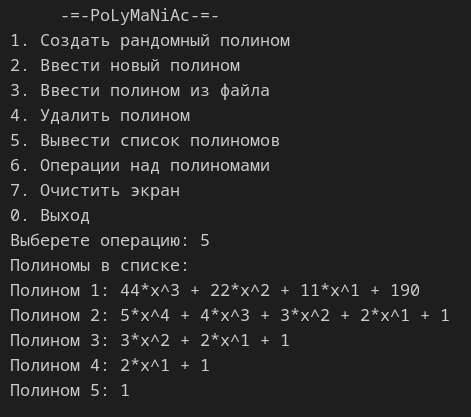


Рис 10. «Вывод актуального списка полиномов после удаления»

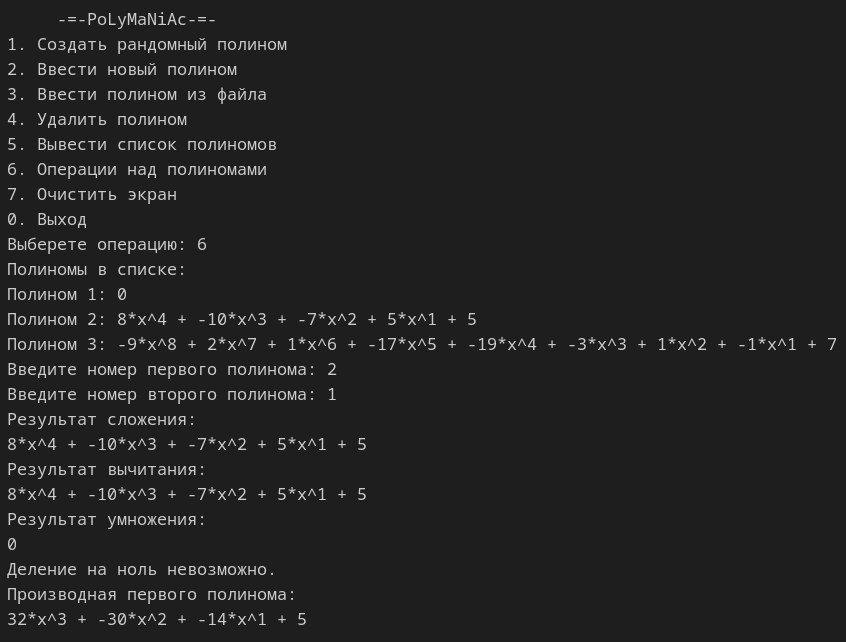
1. В шестом пункте производится выбор двух полиномов (При введении некорректных номеров обрабатывается ошибка) из списка и вывод результатов в операций над ними Рис 11 и 12. 

Рис 11. «Операции с делением на ноль»

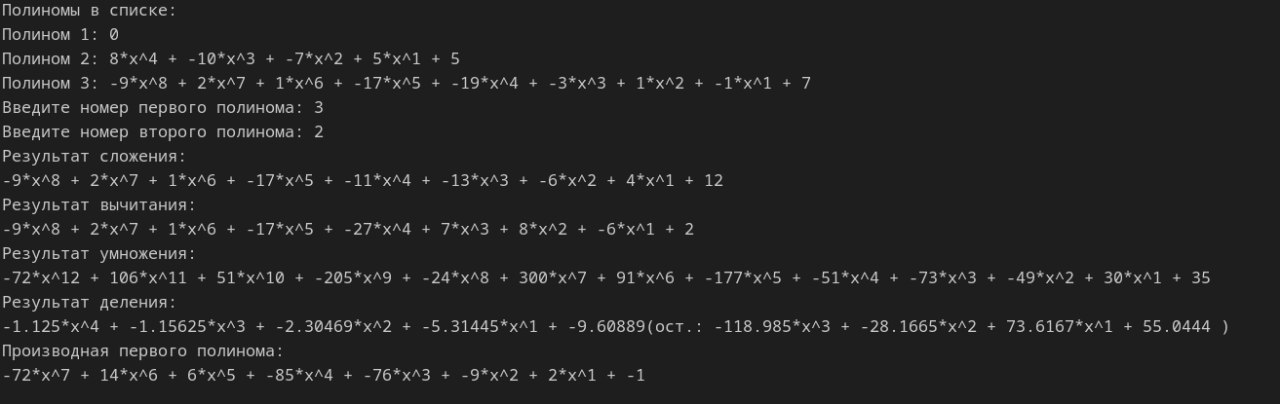


Рис 12. «Операции без деления на ноль»