МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ОБРАЗОВАНИЯ

«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

*Системы Сбора И Обработки Данных*

Кафедра \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(полное название кафедры)

**Курсовая работа**

по дисциплине: Програмирование

наименование дисциплины в соответствии с УП

Тема: Разрбаотка класса Полиндром

Рецензия: (краткое обоснование оценки):

Выполнил:

Студент:Жуков И.А.

Группа: АТ-34

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

подпись

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_ г.

Проверил:

Преподаватель: Воронов В.В.

Балл:\_\_\_\_\_\_\_\_\_\_, ECTS\_\_\_\_\_\_\_\_\_\_\_\_,

Оценка\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Подпись

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.

Новосибирск 2024

Содержание  
  
Технической задание — 1стр  
Описание разработанного класса — 1 стр

* **Техническое задание**

Разработать класс, представляющий заданный тип данных (ТД). Класс должен включать в себя (если это необходимо и подходит по смыслу к ТД):

* Конструкторы (по умолчанию, с параметрами, копии);
* Деструктор;
* Функцию вывода на экран;
* Перегруженные операторы:
  + Оператор присваивания;
  + Бинарные арифметические (+, -, \*, /) с операндами ТД и ТД, ТД и базовый тип, базовый тип и ТД;
  + Арифметика с накоплением (+=,-=);
  + Унарные (++,--) в префиксной и постфиксной форме;
  + Логические (<, >, ==, != );
  + Операторы взятия элемента ( [] или () ) по номеру или ключу;
  + Операторы преобразования типа к любому базовому.

Написать головную программу, обеспечивающую проверку всех реализованных свойств и методов класса.

В варианте 10 нужно реализовать полином

* **Описание разработаного класса**

Polynomial — это класс, представляющий полином с целочисленными коэффициентами, который поддерживает базовые арифметические операции, операции сравнения, индексный доступ к коэффициентам и различные дополнительные операции. Класс позволяет создавать полиномы из списка коэффициентов, выполнять математические операции между полиномами, получать значение полинома при заданном значении переменной и производить преобразования типа.

### Члены класса

#### Конструкторы и деструкторы

* Polynomial() — Конструктор по умолчанию. Создает полином, представляющий собой ноль.
* Polynomial(const std::vector<int>& coeffs) — Конструктор, принимающий вектор коэффициентов coeffs, где каждый элемент представляет коэффициент при соответствующей степени переменной.
* Polynomial(const Polynomial& other) — Конструктор копирования. Создает полином, копируя коэффициенты из other.
* ~Polynomial() — Деструктор. Очищает ресурсы, если это необходимо (в данном случае ресурсы std::vector очищаются автоматически).

#### Операторы присваивания и преобразования

* Polynomial& operator=(const Polynomial& other) — Оператор присваивания, копирующий содержимое other в текущий полином.
* operator int() const — Преобразование в int, возвращающее старший коэффициент полинома.
* operator double() const — Преобразование в double, возвращающее сумму всех коэффициентов (эквивалентно значению полинома при x = 1).

#### Арифметические операторы

* Polynomial operator+(const Polynomial& other) const — Оператор сложения. Возвращает сумму текущего полинома и other.
* Polynomial operator-(const Polynomial& other) const — Оператор вычитания. Возвращает разность текущего полинома и other.
* Polynomial operator\*(const Polynomial& other) const — Оператор умножения. Возвращает результат умножения текущего полинома на other.
* Polynomial operator\*(int scalar) const — Умножение на скаляр. Возвращает полином, полученный умножением текущего полинома на целое число scalar.
* Polynomial operator/(int scalar) const — Деление на скаляр. Возвращает полином, полученный делением текущего полинома на целое число scalar.

#### Операторы с накоплением

* Polynomial& operator+=(const Polynomial& other) — Оператор +=. Складывает other с текущим полиномом.
* Polynomial& operator-=(const Polynomial& other) — Оператор -=. Вычитает other из текущего полинома.

#### Унарные операторы

* Polynomial& operator++() — Префиксный инкремент. Увеличивает свободный член полинома на 1.
* Polynomial operator++(int) — Постфиксный инкремент. Увеличивает свободный член полинома на 1 и возвращает состояние полинома до увеличения.
* Polynomial& operator--() — Префиксный декремент. Уменьшает свободный член полинома на 1.
* Polynomial operator--(int) — Постфиксный декремент. Уменьшает свободный член полинома на 1 и возвращает состояние полинома до уменьшения.

#### Логические операторы

* bool operator==(const Polynomial& other) const — Возвращает true, если текущий полином равен other.
* bool operator!=(const Polynomial& other) const — Возвращает true, если текущий полином не равен other.
* bool operator<(const Polynomial& other) const — Возвращает true, если текущий полином имеет степень меньше, чем у other.
* bool operator>(const Polynomial& other) const — Возвращает true, если текущий полином имеет степень больше, чем у other.

#### Операторы доступа

* int operator[](size\_t index) const — Возвращает коэффициент при степени index. Бросает исключение std::out\_of\_range, если индекс выходит за границы.
* int& operator[](size\_t index) — Возвращает ссылку на коэффициент при степени index. Бросает исключение std::out\_of\_range, если индекс выходит за границы.

#### Вспомогательные функции

* void print() const — Выводит полином в удобочитаемом виде.

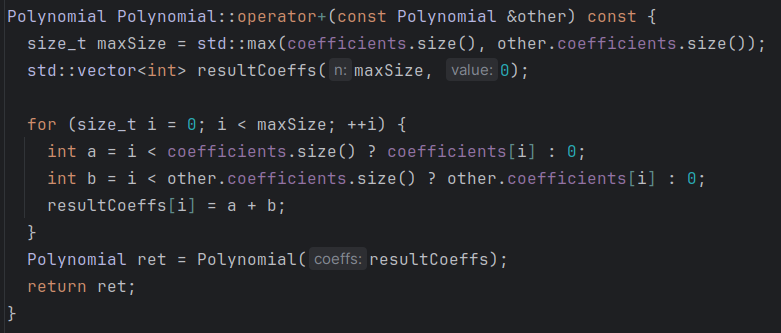
int evaluate(int x) const — Возвращает значение полинома при заданном x.

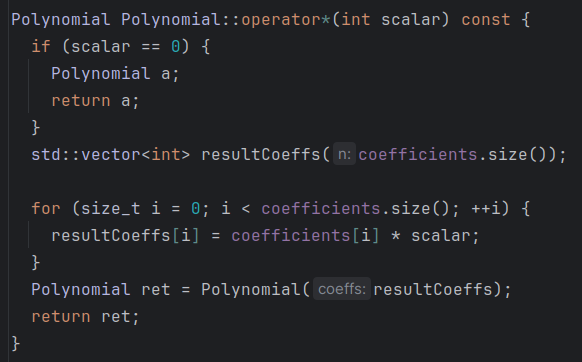
Разработать класс, представляющий заданный тип данных (ТД). Класс должен включать в себя (если это необходимо и подходит по смыслу к ТД):

* Конструкторы (по умолчанию, с параметрами, копии);
* Деструктор;
* Функцию вывода на экран;
* Перегруженные операторы:

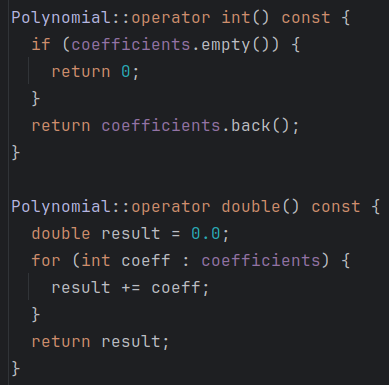
Приложения

Основные арифметические операции





Операторы преобразования типа



Оператор индексации

