**МИНОБРНАУКИ РОССИИ**
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**
**Кафедра Информационных систем**

# ОТЧЕТ

**по лабораторной работе №1**

**по дисциплине «Объектно-ориентированное программирование»**

Студенты гр. 9374 _____ Богданова Е.М.
Калиненкова М.В.

Преподаватель _____ Егоров С.С.

Санкт-Петербург
2022

# ЗАДАНИЕ

Создать консольное приложение согласно представленной на рис.1 диаграмме классов, предназначенное для вычисления корней полинома 2-ой степени p(x) = a*x2 + b*x + c (a≠0) и его значения для заданного аргумента x на множестве целых чисел. Для этого необходимо специфицировать пользовательские классы "Консольное приложение" и "Полином 2ой степени". Т.е. задать атрибуты и методы указанных классов и распределить их по существующим областям видимости. Спецификация классов и реализация их методов должна обеспечивать реализацию отношений, указанных на диаграмме классов. В отчете представить аргументированное обоснование своего выбора.

Приложение должно включать основной модуль (функция main), модуль «application» и модуль «polinom».

В основном модуле консольного приложения (для языка C++ - это модуль с функцией main) должен создаваться объект класса "Консольное приложение" и вызываться его метод, который предоставляет пользователю меню команд приложения.

Модуль «application» должен содержать спецификацию класса "Консольное приложение" и реализацию его методов. Один из методов должен реализовывать меню команд приложения, включающее:

- команду, инициирующую ввод коэффициентов a, b, c (до ввода должны быть заданы значения по умолчанию);
- команду, инициирующую расчета корней полинома и вывод результатов расчета;
- команду, инициирующую ввод значения аргумента x (по умолчанию равен 0), расчет значения и его вывод;
- команду, инициирующую вывод текстового представления полинома в указанной форме p(x);

- команду, инициирующую вывод текстового представления полинома в канонической форме;
- команду выхода из приложения.

Модуль «polinom» должен содержать спецификацию класса "Полином 2ой степени" и реализацию его методов, необходимых для реализации цели разрабатываемого приложения. Описание класса должно использовать вместо типа int (вещественное число, заданное в условии) абстрактный тип number, описание которого должно задаваться в отдельном заголовочном файле number.h с помощью оператора typedef int number (для C++).

Требуется реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

## СПЕЦИФИКАЦИИ КЛАССОВ

1. ConsoleApplication - класс для связи пользователя и других классов между собой.

| | |
|---|---|
| private  Scanner scanner | Объект класса Scanner , используется для ввода из консоли. |
| private Polinom polinom | Представление полинома. |
| private CustomType polinomType | Тип полинома(пока только целочисленный). |
| private boolean exit = false | Переменная, используемая для проверки, нажата ли кнопка выход. |
| public ConsoleApplication() | Конструктор класса, в нем инициализируется объект типа Scanner. |
| public void exec() | Метод для запуска приложения. |
| private void chooseType() | Метод для выбора типа полинома. |
| private int getNumber() | Метод для проверки ввода числа. |
| private void menu() | Главный метод класса, который отвечает за обработку команд ввода. |
| private void createPolinom() | Метод для ввода значений коэффициентов и создания полинома. |
| private void printOptionsMenu() | Метод печатает к консоли возможные операции. |

2. Polinom – класс для представления в программе полинома второй степени.

| | |
|---|---|
| private T a | Переменная, хранящая значение коэффициента при x^2. |
| private T b | Переменная, хранящая значение |

| | |
|---|---|
| | коэффициента при x. |
| private T c | Переменная, хранящая значение свободного члена. |
| private T x1 | Переменная, хранящая значение первого корня. |
| private T x2 | Переменная, хранящая значение второго корня. |
| private T result | Переменная, хранящая результат вычисления с заданным x. |
| public Polinom(T a, T b, T c) | Конструктор класса, принимает коэффициенты. |
| public boolean getRoots() | Метод, вычисляющий корни полинома. Если корни вычислить удалось, возвращает true, если нет, возвращает false и сообщает пользователю, что корней нет. |
| public void printEquation() | Метод выводит в консоль полином в виде a*x^2+b*x+c. |
| public void printCanonical() | Метод выводит в консоль полином в виде a(x-x1)(x-x2). |
| public T solveWithX(CustomNumber x) | Метод вычисляет значение полинома в заданной точке x. |
| public void printRoots() | Метод выводит в консоль корни полинома. |

3. Main – главный класс, содержит метод main, который является точкой входа.

| | |
|---|---|
| public static void main(String[] args) | Точка входа в программу, где создается объект класса |

| | |
|---|---|
| | ConsoleApplication и вызывается его функция exec(). |

4. CustomNumber – абстрактный класс, представляющий число.

| | |
|---|---|
| public abstract CustomNumber plus(CustomNumber a) | Абстрактный метод для сложения. |
| public abstract CustomNumber minus(CustomNumber a) | Абстрактный метод для вычитания. |
| public abstract CustomNumber mult(CustomNumber a) | Абстрактный метод для умножения. |
| public abstract CustomNumber div(CustomNumber a) | Абстрактный метод для деления. |
| public abstract boolean biggerZero() | Абстрактный метод для проверки больше ли нуля число. |
| public abstract boolean isZero() | Абстрактный метод для проверки является ли число нулем. |
| public abstract CustomNumber sqrt() | Абстрактный метод для вычисления квадратного корня числа. |

5. CustomInteger – класс для представления целых чисел, наследуется от CustomNumber.

| | |
|---|---|
| private int value | Переменная, хранящая значение числа. |
| public CustomInteger(int value) | Конструктор класса, принимает значение. |
| private CustomInteger convertToCustomInteger(CustomNumber a) | Метод кастует CustomNumber к CustomInteger. |
| @Override<br>public CustomInteger plus(CustomNumber a) | Метод для сложения CustomInteger. |
| @Override<br>public CustomInteger minus(CustomNumber a) | Метод для вычитания CustomInteger. |
| @Override<br>public CustomInteger | Метод для умножения |

| | |
|---|---|
| mult(CustomNumber a) | CustomInteger. |
| @Override<br>public CustomInteger<br>div(CustomNumber a) | Метод для деления CustomInteger. |
| @Override<br>public boolean biggerZero() | Метод для проверки больше ли CustomInteger нуля. |
| @Override<br>public boolean isZero() | Метод для проверки, равен ли CustomInteger нулю. |
| @Override<br>public CustomNumber sqrt() | Метод для вычисления квадратного корня CustomInteger. |
| @Override<br>public String toString() | Метод для описания вида вывода в консоль CustomInteger. |

6. CustomType – перечисление, содержит все возможные типы полинома.

| | |
|---|---|
| CUSTOM_INT | Константа, соответствующая целочисленному типу. |

Для выполнения данной работы был выбран высокоуровневый язык программирования Java.

Основным классом данной работы является класс ConsoleApplication. Он отвечает за:

1. Создание объекта класса Scanner, который, в свою очередь, отвечает за весь ввод пользователем данных.
2. Получение значения введённого полинома второй степени и создания на его основе экземпляра класса Polinom.
3. Запуск методов для нахождения корней и значения функции у созданного объекта класса Polinom.
4. Получение результатов методов, описанных в пункте выше и выводе их на экран консоли.

При запуске приложения пользователю предлагается выбрать тип полинома, на данный момент доступен только целочисленный. Далее создается полином (по умолчанию с коэффициентами a=1, b=0, c=0) и выводится меню с возможными действиями.

Вторым созданным классом является класс Polinom. Данный класс является представлением полинома второй степени в коде со всеми необходимыми методами, которые необходимы по заданию.
Данный класс имеет публичный конструктор с тремя обязательными параметрами типа T, который должен быть наследником класса CustomNumber.

Для того, чтобы получить корни уравнения, используется публичный метод getRoots(), который не принимает аргументом, т.к. корни ищутся при уравнении, равном 0.

В данном методе вычисляется дискриминант, а также выполняется проверка: если дискриминант меньше нуля, то действительных корней не существует, что, в итоге, и будет выведено на экран. В противном случае корни будут вычислены.

Для вывода в консоль используется метод printRoots(), который просто печатает корни полинома.

Есть два варианта вывода самого полинома: в форме a*x^2+b*x+c и в канонической форме a(x-x1)(x-x2), за это отвечают методы printEquation() и printCanonical() соответственно. В первом просто записываем все коэффициенты, иксы и знаки в одну строку, которую после выводим на экран. Во втором – проверяем, вычислены ли корни, если они не вычислены, сообщаем об этом пользователю, если вычислены, соединяем все в одну строку и выводим ее в консоль.

Для поиска значения полинома в конкретной точке, используется метод solveWithX(), который принимает x.

Для возможности создания полиномов разных типов и переопределения арифметических операций используется абстрактный класс CustomNumber. В этой работе требовалось реализовать работу с целочисленным типом, поэтому был создан класс CustomInteger, он унаследован от CustomNumber. Этот класс является оберткой над обычным интом и нужен только, чтобы

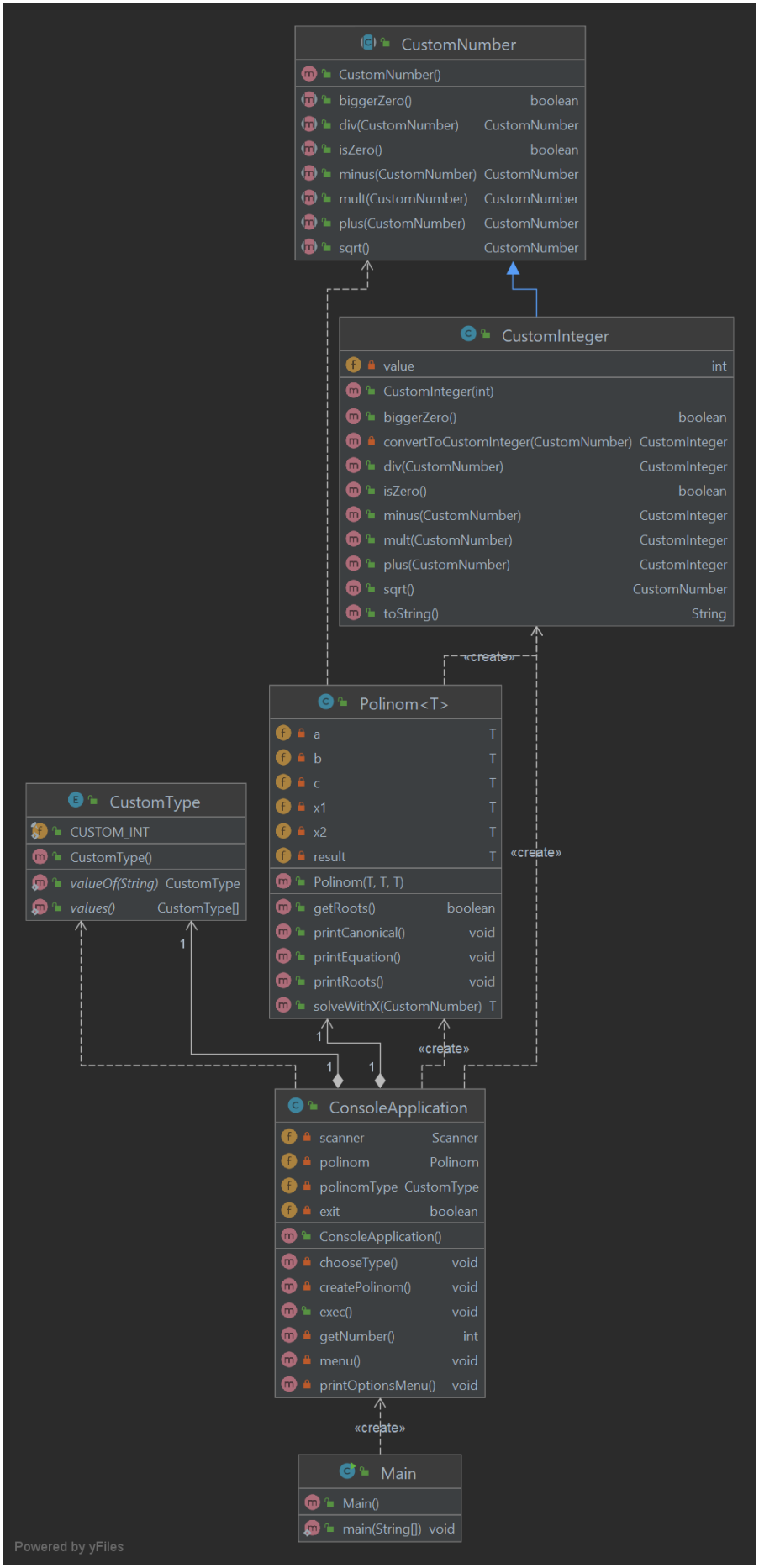
переопределить арифметические операции и вывод в консоль(метод toString()).

Перечисление CustomType пока содержит только одно значение CUSTOM_INT, обозначающее целочисленный тип. Этот тип будет нужен в дальнейших работах, чтобы облегчить проверку типа введенного коэффициента.

Класс Main содержит статический метод main(), который является точкой входа в программу. В нем мы создаем объект типа ConsoleApplication и вызываем его метод exec().

Язык Java был выбран потому, что у него очень удобный функционал работы с классами (в отличие от C++), при максимальной приближённости синтаксиса к C++ по сравнению с аналогами (Python, например).

# ДИАГРАММА КЛАССОВ



## ПРИМЕРЫ РАБОТЫ ПРИЛОЖЕНИЯ

Пример 1.
Решим уравнение 100*x^2-200*x-300=0
И вычислим значение при x=10

```
a = 100
b = -200
c = -300
1.Ввести коэффициенты a, b, c
2.Рассчитать корни и вывести в консоль
3.Ввести x и рассчитать значения полинома
4.Текстовое представление полинома в форме a*x^2 + b*x + c
5.Вывод тестового представления полинома в канонической форме a(x - x1)(x - x2)
6.Выход
2
x1 = 3
x2 = -1
```

```
1.Ввести коэффициенты a, b, c
2.Рассчитать корни и вывести в консоль
3.Ввести x и рассчитать значения полинома
4.Текстовое представление полинома в форме a*x^2 + b*x + c
5.Вывод тестового представления полинома в канонической форме a(x - x1)(x - x2)
6.Выход
3
При x = 0: -300
x = 10
При x = 10: 7700
```

Результат работы программы совпадает с ожидаемым.

Пример 2.

Решим уравнение 4*x^2+5*x-6=0
И вычислим значение при x = 25

```
a = 4
b = 5
c = -6
1.Ввести коэффициенты a, b, c
2.Рассчитать корни и вывести в консоль
3.Ввести x и рассчитать значения полинома
4.Текстовое представление полинома в форме a*x^2 + b*x + c
5.Вывод тестового представления полинома в канонической форме a(x - x1)(x - x2)
6.Выход
2
x1 = 0
x2 = -2
```

```
1.Ввести коэффициенты a, b, c
2.Рассчитать корни и вывести в консоль
3.Ввести x и рассчитать значения полинома
4.Текстовое представление полинома в форме a*x^2 + b*x + c
5.Вывод тестового представления полинома в канонической форме a(x - x1)(x - x2)
6.Выход
3
При x = 0: -6
x = 25
При x = 25: 2619
```

Результат работы совпадает с ожидаемым.

# ВЫВОД

В данной практической работе были разобраны основы объектно-ориентированного программирования, инициализация, вызов методов, присвоение переменных, разграничение доступа к полям и методам, скоупы классов, создание статических и динамических полей, констант, внутренних классов.