

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

ОТЧЕТ
по практической работе №2
по дисциплине «Объектно-ориентированное программирование»

Студент гр. 8374	_____	Пихтовников К.С.
Студент гр. 8374	_____	Подсекин Г.С.
Преподаватель	_____	Егоров С.С.

Санкт-Петербург

2021

Задание на практическую работу

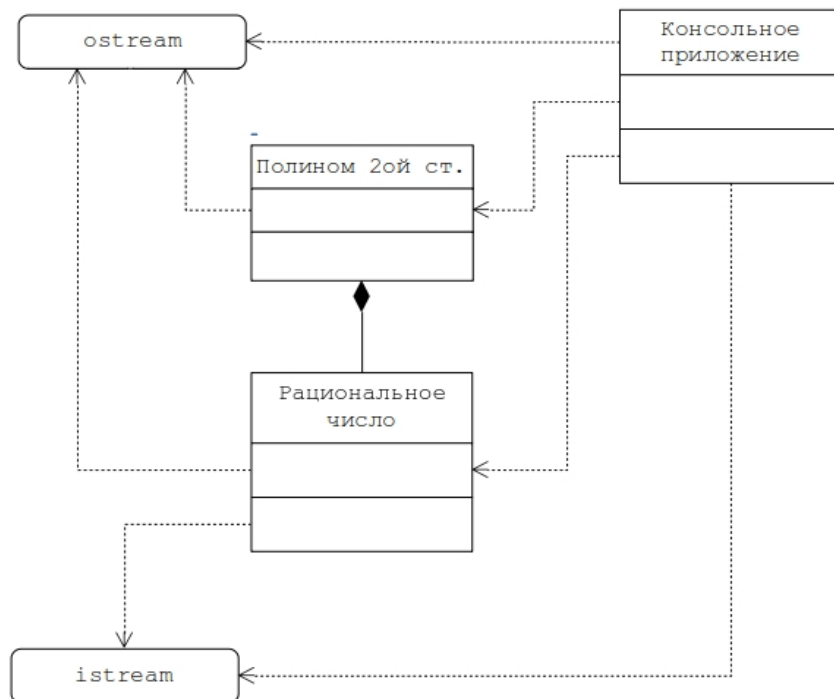


Рис.1. Диаграмма классов работы №2

Создать консольное приложение, реализующее функции перечисленные в описании работы №1 (вычисление корней, вычисление значения, представление полинома в классической и канонических формах) на множестве рациональных чисел.

Приложение должно включать основной модуль, модуль «application», модуль «polinom» и модуль «rational».

Для этого в проект лабораторной работы №1 следует добавить модуль с описанием и реализацией класса рациональных чисел TRational. Класс TRational должен быть встроен в проект согласно диаграмме классов на рис.1. При этом основной модуль, модуль «application» и модуль «polinom» не должны изменяться.

В классе **TRational** следует определить только те члены класса и спецификации, которые необходимы для совместимости модулей проекта и реализации отношений, приведенных в ДК объектной модели.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленной цели. Разработать контрольные примеры и протестировать на них программу. Оформить отчет, сделать выводы по работе.

Спецификация классов

Класс Tapplication

Предназначен для выполнения функций ввода коэффициентов полинома, значения аргумента, инициализации процесса вычисления корней, инициализации процесса вычисления и вывода полинома в классической и канонической формах.

Метод/атрибут	Описание
Метод exes()	Формальных параметров нет, тип void, область видимости-public. В этом методе идет вызов функции menu(), задается конкретное действие, которое пожелал сделать пользователь и результат выводится на экран.
Метод menu()	Формальных параметров нет, тип возвращаемого значения-int, область видимости-private. Вывод на экран необходимого меню, с помощью которого пользователь взаимодействует с программой.
Метод Tapplication()	Конструктор класса

Таблица 1. Класс Tapplication

Класс Tpolinom

Методы и атрибуты данного класса необходимы для выполнения цели разрабатываемой программы (например, получение значений коэффициентов полинома, вычисление дискриминанта, вывод уравнения на экран).

Метод/атрибут	Описание
Атрибут number a, b, c	область видимости – private, хранит значение коэффициентов. По умолчанию коэффициенты равны: a=1, b=2, c=1.
Атрибут EPrintMode printMode	область видимости – private, содержит вид уравнения для печати, который выбрал пользователь
Метод TPolinom(number,number,number)	Конструктор класса
Метод Void setPrintMethod(EPrintMode ePrintMethod)	Тип формального параметра - EPrintMode, область видимости public. Метод устанавливает вид полинома, в котором его необходимо вывести (классический или канонический)
Методы getA(), getB(), getC()	Формальных параметров нет, тип number, область видимости public.

	Предназначены для получения коэффициентов a, b, c.
Метод <code>Int QuantityOfRoots()</code>	Формальных параметров нет, тип возвращаемого значения - <code>int</code> , область видимости <code>public</code> . Возвращает количество корней полинома
Метод <code>number getValue(number x)</code>	Тип формального параметра - <code>number</code> , тип возвращаемого значения - <code>number</code> , область видимости <code>public</code> . Метод вычисляет и возвращает значение полинома.
Метод <code>number Discriminant()</code>	Формальных параметров нет, тип возвращаемого значения - <code>number</code> , область видимости <code>private</code> . Возвращает значение дискриминанта
Метод <code>Bool RootsInteger (number*ArrayOfRoots, int quantityRoots, number a, number b, number c)</code>	Тип возвращаемого значения - <code>bool</code> , область видимости <code>private</code> . Типы формальных параметров: указатель на массив с корнями (<code>number*</code>), <code>int</code> количество корней, <code>number</code> коэффициенты a, b, c. Метод позволяет определить, являются ли корни уравнения целыми числами.
Метод <code>number *Roots()</code>	Формальных параметров нет, тип возвращаемого значения - <code>number</code> , область видимости <code>public</code> . Возвращает указатель на массив с корнями.
Метод <code>ostream& operator << (ostream& os, TPolinom& p)</code>	Тип возвращаемого значения – <code>ostream object</code> , область видимости – <code>public</code> . Данный метод выводит на экран полином в классической или канонической форме.

Таблица 2. Класс *TPolinom*

Класс TRational

Методы и атрибуты данного класса необходимы для вычисления корней, значений, представления полинома в классической и канонических формах на множестве рациональных чисел.

Метод/атрибут	Описание
Атрибут <code>int numerator</code>	область видимости - <code>private</code> . Является числителем дроби (целое число)
Атрибут <code>unsigned int denominator</code>	область видимости - <code>private</code> . Является знаменателем дроби (целое число, большее нуля)
Метод <code>TRational()</code>	Конструктор класса
Метод <code>TRational(const int&)</code>	Конструктор класса

Метод unsigned int NOK(const TRational&)	Тип формального параметра - const TRational&, тип возвращаемого значения - unsigned int, область видимости private. Предназначен для вычисления наименьшего общего кратного.
Метод int NOD(const int&,const int&)	Тип формального параметра - const int&, тип возвращаемого значения - int, область видимости private. Предназначен для вычисления наибольшего общего кратного.
Метод void decrease()	Формальных параметров нет, область видимости private. Метод позволяет привести дробь к несократимому виду
Метод TRational operator+ (const TRational&)	Тип формального параметра - const TRational&, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка оператора сложения
Метод TRational operator* (const TRational&)	Тип формального параметра - const TRational&, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка оператора умножения
Метод TRational operator* (const int&)	Тип формального параметра - const int&, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка оператора умножения на число
Метод TRational operator/ (const TRational&)	Тип формального параметра - const TRational&, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка оператора деления
Методы: bool operator > (int) bool operator < (int) bool operator == (int)	Тип формального параметра - int, тип возвращаемого значения - bool, область видимости public. Перегрузка операторов сравнения (для целых чисел)
Метод bool operator == (TRational)	Тип формального параметра - TRational, тип возвращаемого значения - bool, область видимости public. Перегрузка оператора сравнения (проверка сравнения дробей)
Метод TRational operator- (const TRational&)	Тип формального параметра - const TRational&, тип возвращаемого значения - объект класса TRational,

	<p>область видимости public.</p> <p>Перегрузка бинарного минуса</p>
Метод TRational operator- ()	<p>Формальных параметров нет,</p> <p>тип возвращаемого значения - объект класса TRational,</p> <p>область видимости public.</p> <p>Перегрузка унарного минуса</p>
Метод friend TRational sqrt(TRational)	<p>Тип формального параметра - TRational, тип возвращаемого значения - объект класса TRational, область видимости public.</p> <p>Перегрузка оператора, который вычисляет квадратный корень</p>
Метод friend ostream& operator<< (ostream& os, const TRational&)	<p>Тип формальных параметров- ostream& os, const TRational&, тип возвращаемого значения - ostream object, область видимости – public.</p> <p>Перегрузка оператора вывода</p>
Метод friend istream& operator>> (istream& is, TRational&)	<p>Тип формальных параметров - istream& is, TRational&, тип возвращаемого значения – istream object, область видимости – public.</p> <p>Перегрузка оператора ввода</p>

Таблица 3. Класс TRational

Диаграмма классов

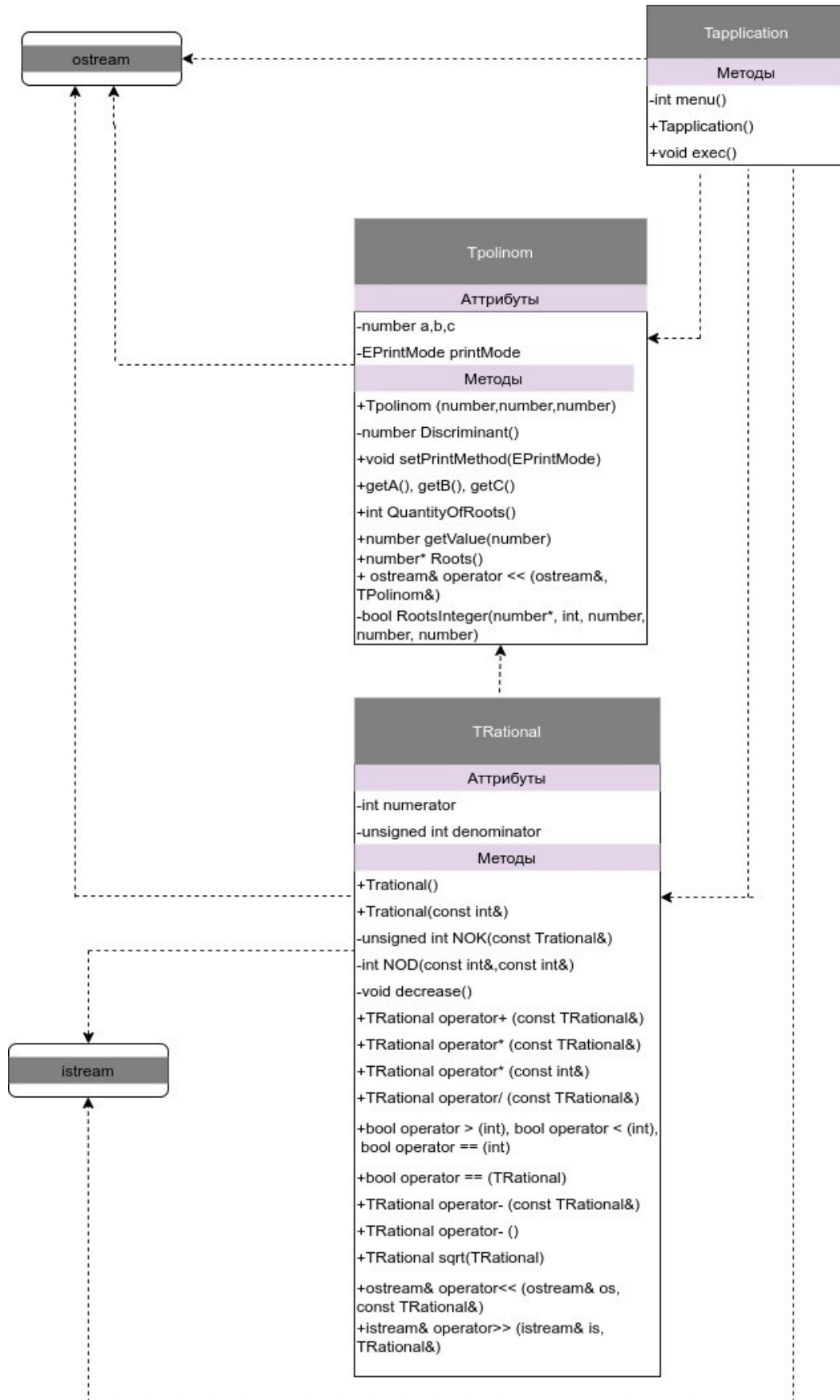


Рис.2. Реализация диаграммы классов работы №2

Символ	Значение
+	public - открытый доступ
-	private - только из операций того же класса
#	protected - только из операций этого же класса и классов, создаваемых на его основе

Таблица 4. Обозначение атрибутов и методов класса

Описание контрольного примера с исходными и ожидаемыми (расчетными) данными

Пример 1:

Исходные данные:

Коэффициенты:

$$a = \frac{4}{8}; b = \frac{9}{12}; c = -\frac{16}{32};$$

Ожидаемые данные:

$$x_1 = -2, x_2 = 0.5$$

$$p(1/6) = \left(\frac{4}{8}\right) * \left(\frac{1}{6}\right)^2 + \left(\frac{9}{12}\right) * \left(\frac{1}{6}\right) - \left(\frac{16}{32}\right) = -\frac{13}{36}$$

$$\text{Классический вид: } \frac{4}{8}x^2 + \frac{9}{12}x - \frac{16}{32}$$

$$\text{Канонический вид: } \frac{4}{8} * \left(x - \frac{1}{2}\right) * (x + 2)$$

Пример 2:

Исходные данные:

Коэффициенты:

$$a = \frac{1}{2}, b = \frac{1}{2}, c = \frac{1}{8}$$

Ожидаемые данные:

$$x_1 = x_2 = -0.5$$

$$p(1/3) = \frac{1}{2} * \left(\frac{1}{3}\right)^2 + \left(\frac{1}{2}\right) * \left(\frac{1}{3}\right) + \left(\frac{1}{8}\right) = \frac{25}{72}$$

$$\text{Классический вид: } \frac{1}{2}x^2 + \frac{1}{2}x + \frac{1}{8}$$

$$\text{Канонический вид: } \frac{1}{2} * \left(x + \frac{1}{2}\right)^2$$

Пример 3:

Исходные данные:

Коэффициенты:

$$a = \frac{2}{5}; b = \frac{1}{3}; c = \frac{5}{6};$$

Ожидаемые данные:

Корней нет

$$p\left(\frac{2}{5}\right) = \left(\frac{2}{5}\right) * \left(\frac{2}{5}\right)^2 + \left(\frac{1}{3}\right) * \left(\frac{2}{5}\right) + \left(\frac{5}{6}\right) = \frac{773}{750}$$

$$\text{Классический вид: } \frac{2}{5}x^2 + \frac{1}{3}x + \frac{5}{6}$$

Канонический вид: полином не имеет корней, поэтому его невозможно вывести в канонической форме.

Скриншоты программы на контрольных примерах

Пример 1:

```
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>1
Enter a,b,c:
a= 4 8
b= 9 12
c= -16 32
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>1 6
P((1/6))=-(13/36)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
There are two roots: x1=(1/2)   x2=-2
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
(4/8)x^2+(9/12)x-(16/32)
1- coefficients
```

```
>4
(4/8)x^2+(9/12)x-(16/32)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
(4/8)*(x-(1/2))*(x+2)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>
```

Пример 2:

```
Enter a,b,c:
a= 1 2
b= 1 2
c= 1 8
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>1 3
P((1/3))=(25/72)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>3
There is one root: x=-(1/2)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
(1/2)x^2+(1/2)x+(1/8)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
(1/2)*(x+(1/2))^2
```

Пример 3:

```
>1
Enter a,b,c:
a= 2 5
b= 1 3
c= 5 6
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>2
Enter x:
>2 5
P((2/5))=(773/750)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>4
(2/5)x^2+(1/3)x+(5/6)
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>5
The polynomial has no roots, so it is impossible to derive in canonical form
1- coefficients
2- value
3- roots
4- print (classic)
5- print (canonical)
0- exit>
>
```

Вывод

В ходе данной лабораторной работы было создано консольное приложение согласно представленной на рис.1 диаграмме классов. Модули «application», «polinom» и основной модуль (функция main) были задействованы из 1 практической работы и не подвергались изменениям. Также был введен новый модуль «rational», который позволяет решать квадратные уравнения над полем рациональных чисел.

Помимо этого, была создана диаграмма классов (рис.2) и произведена отладка работы программы. Разработаны контрольные примеры с исходными и ожидаемыми данными, которые затем были протестированы в созданном консольном приложении. Все результаты совпали.