

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

ОТЧЕТ
по практической работе №3
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

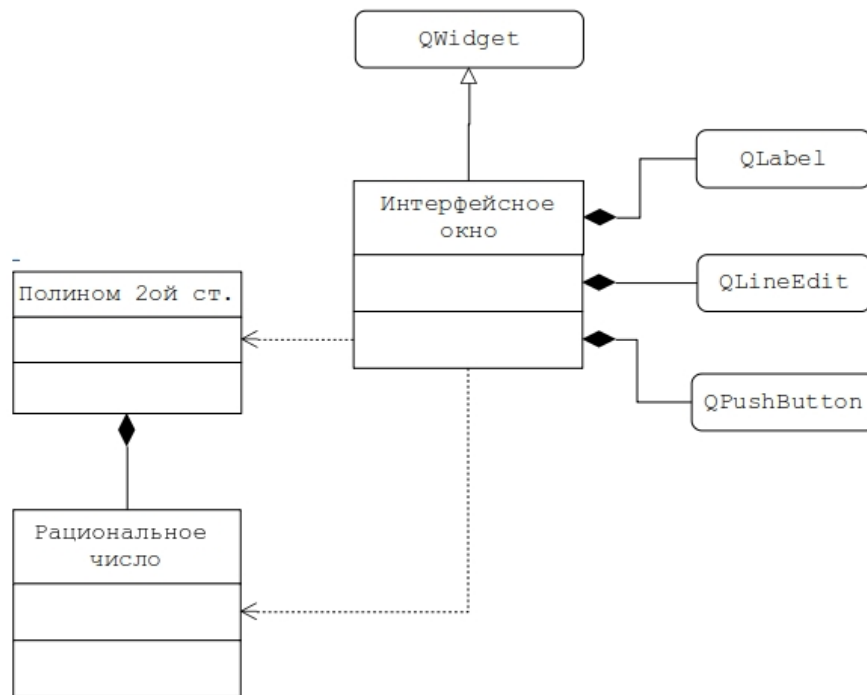


Рис.1. Диаграмма классов работы №3

Создать GUI 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 «interface», модуль «polinom» и модуль «rational».

Основной модуль main.cpp GUI приложения может иметь вид:

```
#include <QApplication>
#include "interface.h"
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    TInterface interface;
    interface.show();
    return a.exec();
}
```

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

Спецификация классов

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 Int QuantityOfRoots()	Формальных параметров нет, тип возвращаемого значения - int, область видимости public. Возвращает количество корней полинома
Метод number getValue(number x)	Тип формального параметра - number, тип возвращаемого значения - number, область видимости public. Метод вычисляет и возвращает значение полинома.
Метод number Discriminant()	Формальных параметров нет, тип возвращаемого значения - number, область видимости private. Возвращает значение дискриминанта
Метод Bool RootsInteger (number*ArrayOfRoots, int quantityRoots, number a, number b, number c)	Тип возвращаемого значения - bool, область видимости private. Типы формальных параметров: указатель на массив с корнями (number*), int количество корней, number коэффициенты a, b, c. Метод позволяет определить, являются ли корни уравнения целыми числами.
Метод number *Roots()	Формальных параметров нет, тип

	возвращаемого значения - number, область видимости public. Возвращает указатель на массив с корнями.
Метод ostream& operator << (ostream& os, TPolinom& p)	Тип возвращаемого значения – ostream object, область видимости – public. Данный метод выводит на экран полином в классической или канонической форме.
Метод friend QString& operator << (QString&, TPolinom&)	Тип возвращаемого значения – QString object, область видимости — public. Является перегруженным оператором, который позволяет перевести тип полинома в строку.

Таблица 1. 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 int numerator	область видимости - private. Является числителем дроби (целое число)
Атрибут unsigned int denominator	область видимости - private. Является знаменателем дроби (целое число, большее нуля)
Метод TRational()	Конструктор класса
Метод TRational(const int&)	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, область видимости public. Перегрузка бинарного минуса
Метод TRational operator- ()	Формальных параметров нет, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка унарного минуса
Метод friend TRational sqrt(TRational)	Тип формального параметра - TRational, тип возвращаемого значения - объект класса TRational, область видимости public. Перегрузка оператора, который вычисляет квадратный корень
Метод friend ostream& operator<< (ostream& os, const TRational&)	Тип формальных параметров- ostream& os, const TRational&, тип возвращаемого значения - ostream object, область видимости – public. Перегрузка оператора вывода
Метод friend istream& operator>> (istream& is,	Тип формальных параметров - istream& is,

is, TRational&)	TRational&, тип возвращаемого значения – istream object, область видимости – public. Перегрузка оператора ввода
Метод TRational(const int&,const int&)	Конструктор класса
Метод friend QString& operator<< (QString& os, const TRational&)	Тип возвращаемого значения – QString object, область видимости — public. Является перегруженным оператором, который позволяет перевести тип полинома в строку.

Таблица 2. Класс Trational

Класс TInterface

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
Атрибуты: QLabel * name_a, *delimiter_a; QLabel * name_b, *delimiter_b; QLabel * name_c, *delimiter_c;	область видимости - private. Являются подсказками, которые показывают пользователю, что нужно вводить
Атрибуты: QLineEdit*a_numerator,*a_denominator; QLabel * name_b, *delimiter_b; QLabel * name_c, *delimiter_c	область видимости - private. Переменные, которые хранят введенные числитель и знаменатель коэффициента a, b и c
Атрибут QPushButton *value_btn	область видимости - private. Кнопка для отображения значения уравнения
Атрибут QPushButton *root_btn	область видимости - private. Кнопка для отображения корней уравнения
Атрибут QLabel *name_x, *delimiter_x	область видимости - private. Являются подсказками, которые показывают пользователю, что нужно вводить значение x
Атрибут QLineEdit *x_numerator, *x_denominator	область видимости - private. Переменные, которые хранят введенные числитель и знаменатель значения x
Атрибут QPushButton *print_classic_btn	область видимости - private. Кнопка для отображения уравнения в классическом виде
Атрибут QPushButton *print_canonic_btn	область видимости - private. Кнопка для отображения уравнения в

	каноническом виде
Атрибут QLabel *output	область видимости - private. Поле, предназначенное для вывода результатов
Метод TInterface(QWidget *parent = nullptr)	Область видимости public. Конструктор класса
Метод ~TInterface()	Область видимости public. Деструктор класса
Метод void roots()	Формальных параметров нет. Область видимости – public Метод предназначен для вычисления корней уравнения
Метод void value()	Формальных параметров нет. Область видимости – public Метод предназначен для вычисления значения уравнения
Метод void printClassic()	Формальных параметров нет. Область видимости – public Метод предназначен для вывода уравнения в классическом виде
Метод void printCanonic()	Формальных параметров нет. Область видимости – public Метод предназначен для вывода уравнения в канонической форме

Таблица 3. Класс TInterface

Диаграмма классов

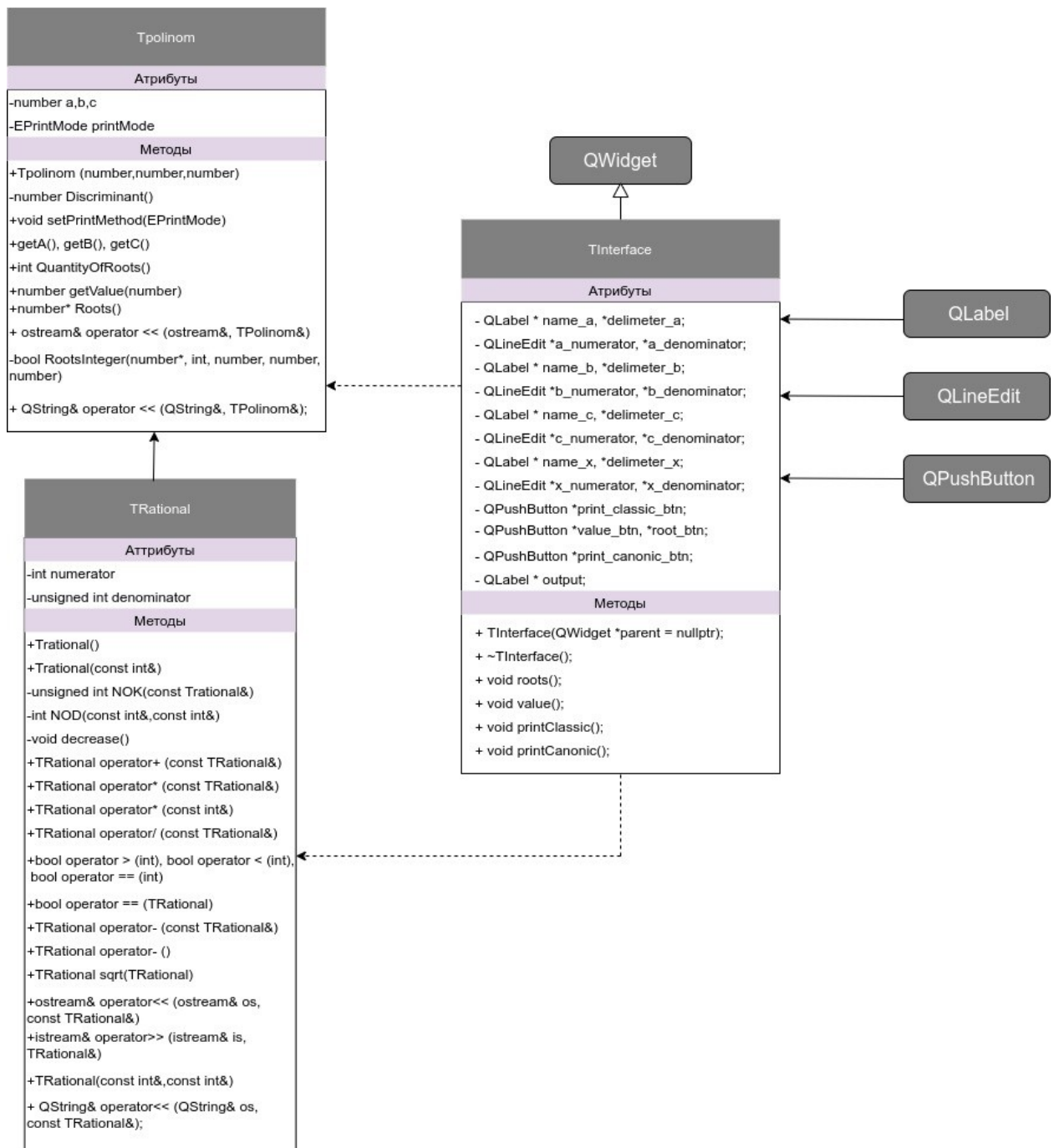


Рис.2. Реализация диаграммы классов работы №2

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:

Лабораторная работа №3

a = /
b = /
c = /
x = /

$p((1/6)) = -(13/36)$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$x_1 = (1/2); x_2 = -2;$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$(4/8) * (x - (1/2)) * (x + 2)$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$(4/8)x^2 + (9/12)x - (16/32)$

Пример 2:

Лабораторная работа №3

a = /
b = /
c = /
x = /

$p((1/3))=(25/72)$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$x = -(1/2);$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$(1/2)*(x+(1/2))^2$

Лабораторная работа №3

a = /
b = /
c = /
x = /

$(1/2)x^2+(1/2)x+(1/8)$

Пример 3:

Лабораторная работа №3

a = /
b = /
c = /
x = /

$p((2/5))=(773/750)$

Лабораторная работа №3

a = /
b = /
c = /
x = /

Корней нет

Лабораторная работа №3

a = /
b = /
c = /
x = /

Канонический вид невозможен

Лабораторная работа №3

a = /
b = /
c = /
x = /

$(2/5)x^2+(1/3)x+(5/6)$

Вывод

В ходе данной лабораторной работы было создано GUI приложение согласно представленной на рис.1 диаграмме классов. Модули «tpolinom» и «rational» были задействованы из 2 практической работы. Также был заново создан основной модуль (функция main) и введен новый модуль «TInterface», который предназначен для выполнения функций ввода коэффициентов полинома, значения аргумента, инициализации процесса вычисления корней, инициализации процесса вычисления и вывода полинома в классической и канонической формах.

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 GUI приложении. Все результаты совпали.