

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

ОТЧЕТ
по практической работе №5
по дисциплине «Объектно-ориентированное программирование»

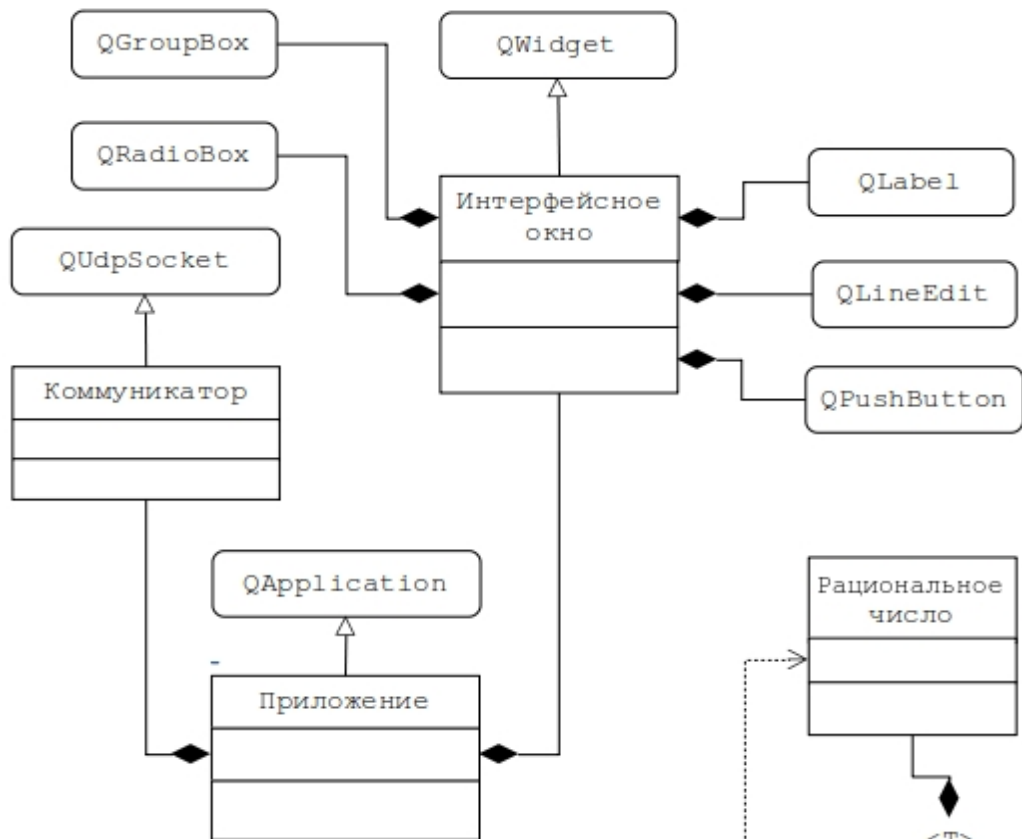
Студент гр. 8374	_____	Пихтовников К.С.
Студент гр. 8374	_____	Подсекин Г.С.
Преподаватель	_____	Егоров С.С.

Санкт-Петербург

2021

Задание на практическую работу

Клиентская часть



Серверная часть

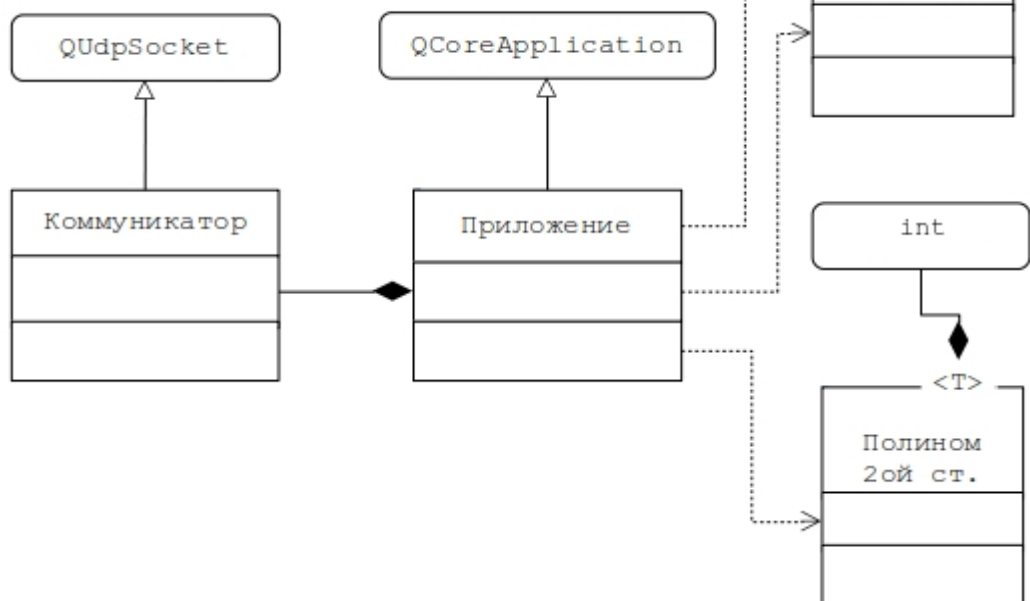


Рис.1. Диаграмма классов работы №5

Создать распределенное приложение, включающее клиентскую и серверную части, взаимодействующие посредством сетевого обмена сообщениями.

Клиентская часть представляет собой GUI приложение, реализующее интерфейс аналогичный работе №3.

Серверная часть представляет собой консольное приложение, предназначенное для вычисления корней, значения и представления полинома 2-ой степени в классической и канонической формах на множествах вещественных и рациональных чисел. Для этого следует модифицировать описание класса **TPolinom**, представив его в виде параметризованного класса с параметром **number**, который может принимать значения как **int**, так и **TRational**.

Указания:

1. Заголовочный файл `number.h` из приложения исключить.
2. Содержимое файла реализации `polinom.cpp` перенести в заголовочный файл `polinom.h` (шаблон класса `TPolinom`) и исключить из проекта.

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

Клиентская часть распределенного приложения:

Класс TInterface

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 answer(QString)	Тип формального параметра - QString, область видимости public. Метод обрабатывает сигнал, поступивший

	из серверной части распределенного приложения
Метод void formRequest()	Формальных параметров нет, область видимости private. Метод формирует запрос с клиентской части приложения.
Метод void request(QString)	Тип формального параметра - QString, область видимости private. Метод преобразует сформированный запрос в сигнал.

Таблица 1. Класс Tinterface

Класс TApplication

Метод/атрибут	Описание
Атрибут TCommunicator *comm	объект класса Tcommunicator, область видимости - private.
Атрибут TInterface *interface	объект класса TInterface, область видимости private
Метод void fromCommunicator(QByteArray)	Тип формального параметра - QByteArray, область видимости public. Метод отображает в интерфейсе результат запроса.
Метод void toCommunicator(QString)	Тип формального параметра - QString, область видимости public. Метод передает сообщение в серверную часть приложения.

Таблица 2. Класс Tapplication

Класс TCommParams

Метод/атрибут	Описание
Атрибут QHostAddress rHost	область видимости — private. Адрес приема
Атрибут quint16 rPort	область видимости — private. Порт приема
Атрибут QHostAddress sHost	область видимости — private. Адрес отправки
Атрибут quint16 sPort	область видимости — private. Порт отправки

Таблица 3. Класс TcommParams

Класс TCommunicator

Метод/атрибут	Описание
Атрибут bool ready	область видимости — private. Принимает 1, если объект класса TCommunicator был успешно привязан к указанному адресу, 0 в обратном случае

Атрибут TCommParams params	область видимости — private. Параметры для подключения.
Метод bool isReady()	Формальных параметров нет, область видимости public. Метод возвращает переменную true типа bool, если ready равно 1, и false, если ready равно 0
Метод void recieved(QByteArray)	Тип формального параметра - QByteArray, область видимости private. Метод преобразует сформированный запрос в сигнал.
Метод void send(QByteArray)	Тип формального параметра - QByteArray, область видимости public. Метод отправляет сформированную датаграмму при наличии соединения между серверной и клиентской частью.
Метод void recieve()	Формальных параметров нет, область видимости private. Метод, при наличии датаграммы, принимает её.

Таблица 4. Класс TcommParams

Серверная часть распределенного приложения:

Класс TPolinom

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 RootsCheck (number*, int, number, number, number)</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 <code>a</code> , <code>b</code> , <code>c</code> . 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.
Метод <code>friend QString& operator << (QString&, TPolinom&)</code>	Тип возвращаемого значения – <code>QString object</code> , область видимости — <code>public</code> . Является перегруженным оператором, который позволяет перевести тип полинома в строку.

Таблица 5. Класс *TPolinom*

Класс *TRational*

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)
Static <code>QChar SEPARATOR</code>	область видимости — <code>private</code> . Является символом разделителя дроби
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,

	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, TRational&)	Тип формальных параметров - istream& is, 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 6. Класс Trational

Класс TApplication

Метод/атрибут	Описание
Атрибут TCommunicator *comm	объект класса Tcommunicator, область видимости - private.
Метод void recieve(QByteArray)	Тип формального параметра - QByteArray, область видимости public. Метод обрабатывает сообщение с запросом от клиентской части распределенного приложения.

Таблица 7. Класс Tapplication

Класс TCommParams

Метод/атрибут	Описание
Атрибут QHostAddress rHost	область видимости — private. Адрес приема
Атрибут quint16 rPort	область видимости — private. Порт приема
Атрибут QHostAddress sHost	область видимости — private.

	Адрес отправки
Атрибут quint16 sPort	область видимости — private. Порт отправки

Таблица 8. Класс TcommParams

Класс TCommunicator

Метод/атрибут	Описание
Атрибут bool ready	область видимости — private. Принимает 1, если объект класса TCommunicator был успешно привязан к указанному адресу, 0 в обратном случае
Атрибут TCommParams params	область видимости — private. Параметры для подключения.
Метод bool isReady()	Формальных параметров нет, область видимости public. Метод возвращает переменную true типа bool, если ready равно 1, и false, если ready равно 0
Метод void recieved(QByteArray)	Тип формального параметра - QByteArray, область видимости private. Метод преобразует сформированный запрос в сигнал.
Метод void send(QByteArray)	Тип формального параметра - QByteArray, область видимости public. Метод отправляет сформированную датаграмму при наличии соединения между серверной и клиентской частью.
Метод void recieve()	Формальных параметров нет, область видимости private. Метод, при наличии датаграммы, принимает её.

Таблица 9. Класс TcommParams

Диаграмма классов

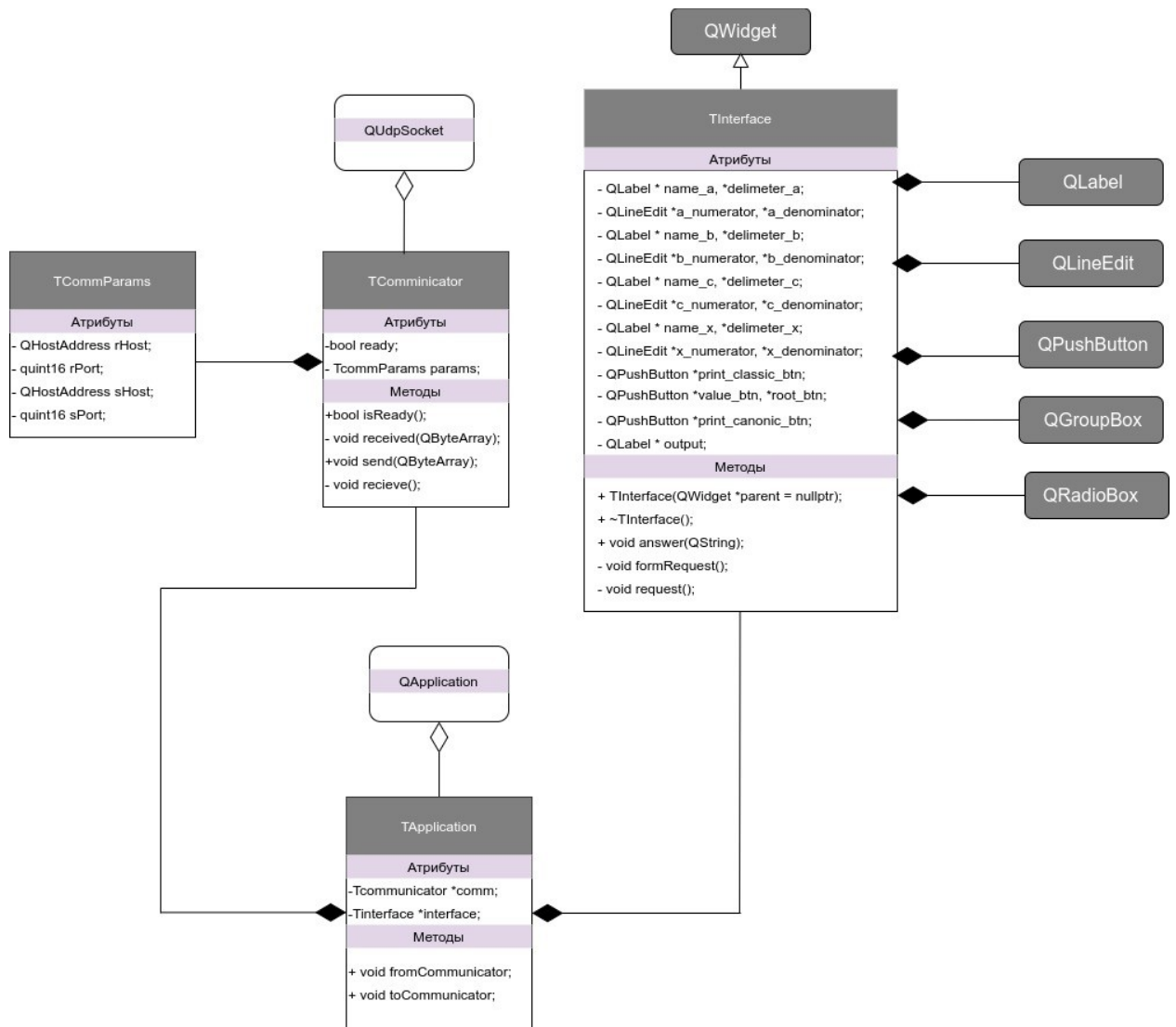


Рис.2. Реализация диаграммы классов клиентской части

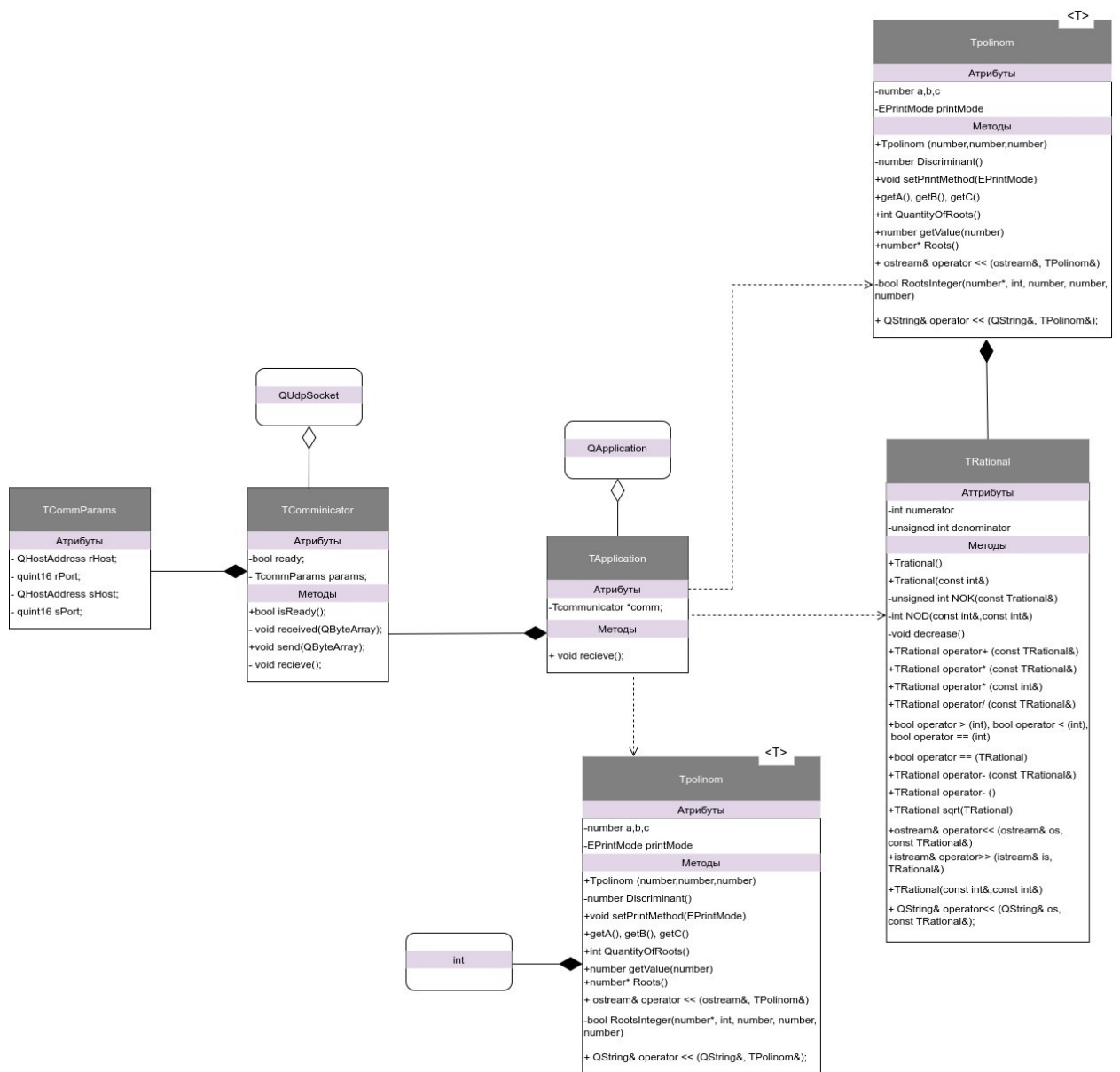


Рис.3. Реализация диаграммы классов серверной части

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 10. Обозначение атрибутов и методов класса

Описание контрольного примера с исходными и ожидаемыми (расчетными) данными

Пример 1(для дробей):

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(для дробей):

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(для целых чисел):

Исходные данные:

Коэффициенты:

$$a = -1, b = 4, c = 0$$

Ожидаемые данные:

$$x_1 = 4, x_2 = 0$$

$$p(1) = -1 * 1 * 1 + 4 * 1 = 3$$

$$\text{Классический вид: } -1x^2 + 4x$$

$$\text{Канонический вид: } -1 * x * (x - 4)$$

Пример 4(для целых чисел):

Исходные данные:

Коэффициенты:

$$a=2, b=4, c=7$$

Ожидаемые данные:

Корней нет

$$p(5)=2*5*5+4*5+7=77$$

Классический вид: $2x^2+4x+7$

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

Скриншоты программы на контрольных примерах

Пример 1(для дробей):

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Лабораторная работа №5

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

☐ Целые ☒ Дроби

$(4/8) \cdot (x - (1/2)) \cdot (x + 2)$

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Пример 2(для дробей):

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Лабораторная работа №5

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

☐ Целые ☒ Дроби

$x = -(1/2)$

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Лабораторная работа №5

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

☐ Целые ☒ Дроби

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

Пример 3(для целых чисел):

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$p(1) = 3$

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$x_1 = 0; x_2 = 4;$

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$-1*x*(x-4)$

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$-1x^2+4x$

Пример 4(для целых чисел):

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$p(5) = 77$

Лабораторная работа №5

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

☒ Целые ☐ Дроби

Корней нет

Лабораторная работа №5

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

☒ Целые ☐ Дроби

Канонический вид невозможен

Лабораторная работа №5

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

☒ Целые ☐ Дроби

$2x^2+4x+7$

Вывод

В ходе данной лабораторной работы было создано распределенное приложение, включающее клиентскую и серверную части, взаимодействующие посредством сетевого обмена сообщениями. Клиентская часть представляет собой GUI приложение, реализующее интерфейс аналогичный работе №3. Серверная часть представляет собой консольное приложение, предназначенное для вычисления корней, значения и представления полинома 2-ой степени в классической и канонической формах на множестве рациональных чисел. Для этого было модифицировано описание класса **TPolinom**, который был представлен в виде параметризованного класса с параметром **number**, который может принимать значения как **int**, так и **TRational**.

Помимо этого, была создана диаграмма классов клиентской части(рис.2) и серверной части(рис.3), а также произведена отладка работы программы. Разработаны контрольные примеры с исходными и ожидаемыми данными, которые затем были протестированы в созданном распределенном приложении. Все результаты совпали.