**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»**

|  |  |  |
| --- | --- | --- |
| Студенты группы № 2372 |  | Полуянов В.Н., Юрин А.А. |
| Преподаватель |  | Егоров С.С. |

Санкт-Петербург

2023

**Задание**

Создать GUI приложение, реализующее функции перечисленные в описании работы №1, но на множестве рациональных чисел. Для этого требуется разработать и реализовать класс рациональных чисел. Рациональное число — это несократимая дробь a / b, где a и b — целые, причем b>0.

Приложение должно включать основной модуль, модуль «interface», модуль «matrix», модуль «rational» и файл number.h.

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 примеры и протестировать на них программу. Оформить отчет, сделать выводы по работе.

**Спецификации классов**

**Class TMatrix**

Атрибуты:

-unsigned short size – определяет размер заданной матрицы

-std::vector<std::vector<number>>values - определяет значения матрицы

Методы:

+TMatrix() – конструктор класса без предустановленного размера

+TMatrix(unsigned short) – конструктор с предустановленным размером

+setValues(std::vector<std::vector<number>>&) – изменяет значения матрицы

+setSize(unsigned short) – изменяет размер матрицы (расширяет или сжимает её с сохранением значений, новые - нули)

+getSize() – возвращает unsigned short текущий размер матрицы

+operator<<(QS&, TMatrix&) – выводит матрицу в QString

+getDeterminant() – возвращает number определитель матрицы

+getRank() – возвращает unsigned int ранг матрицы

+transpose() – транспонирует матрицу

#toUpperTriangularForm() – приводит матрицу к верхнему ступенчатому виду с помощью метода Гаусса. Используется для вычисления определителя и ранга матрицы.

**Class TRational**

Атрибуты:

+numerator – числитель

+divisor – знаменатель

Методы:

+TRational() – стандартный конструктор. Задает число 0

+TRational(double&) – конструктор целого числа

+TRational(double&, double&) – конструктор дробного числа

+operator<< (QString&, TRational) – выводит число в QString

+operator+ (Trational) – перегрузка оператора сложения

+operator- (Trational) – перегрузка оператора вычитания

+operator\* (Trational) – перегрузка оператора умножения

+operator/ (Trational) – перегрузка оператора деления

**Class TInterface**

Атрибуты:

-int matrixSize – размер создаваемой матрицы

-std::vector<QLabel\*> delimeters – вектор полей-разделителей («/»)

-std::vector<std::vector<std::vector<QLineEdit\*>>> numbers – вектор полей ввода значений матрицы

-QPushButton \*print\_btn – кнопка «вывести матрицу»

-QPushButton \*determinant\_btn – кнопка «определитель»

-QPushButton \*rank\_btn – кнопка «ранг»

-QPushButton \*transpose\_btn – кнопка «транспонировать»

Методы:

+TInterface() – конструктор класса

+~TInterface() – деструктор класса

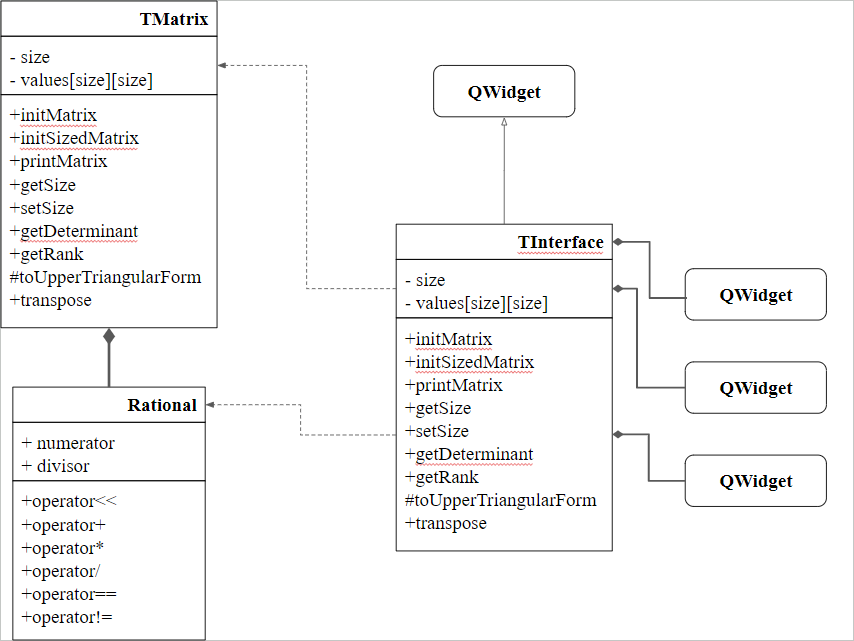
+print() – функция вывода матрицы

+determinant() – функция вывода определителя

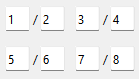
+rank() – функция вывода ранга

+transpose() – функция вывода транспонированной матрицы

**Диаграмма классов, дополненная атрибутами и методами**

****

**Описание контрольного примера с исходными и ожидаемыми (расчетными) данными**

****

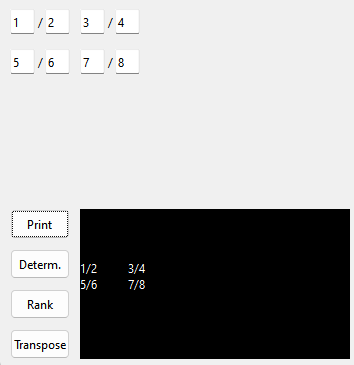
Расчетные результаты следующие:

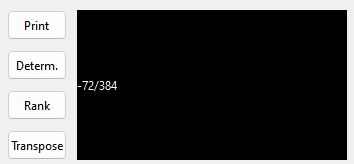
* Определитель равен -0.1875
* Ранг равен 2

При тестировании работы программы были получены идентичные (правильные) результаты.

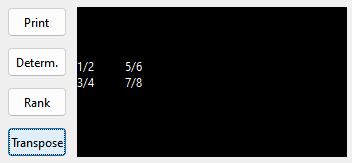
**Скриншоты работы программы на контрольных примерах. Два примера**

1. *Первый пример*



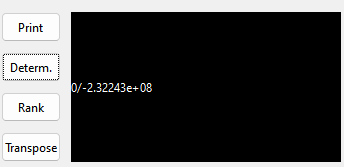


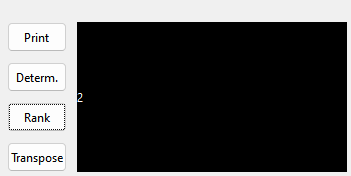


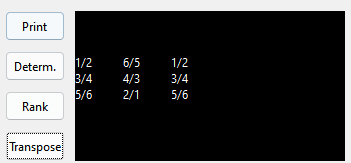


1. *Второй пример*









**Вывод**

В данной практической работе было успешно разработано GUI приложение, реализующее функции перечисленные в описании работы №1, но на множестве рациональных чисел. Это приложение может быть дополнено и расширено для работы с различными типами данных, включая числа из других множеств.