**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информационных систем**

отчет

**по практической работе №1**

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

|  |  |  |
| --- | --- | --- |
| Студенты гр. 0375 |  | Пилипченко С. К.,  Стукалов С. С. |
| Преподаватель |  | Егоров С. С. |

Санкт-Петербург

2023

**Задание на практическую работу:**

Создать консольное приложение согласно представленной диаграмме классов, предназначенное для заданных вычислений над квадратной матрицей, заданной на множестве вещественных чисел. Для этого необходимо специфицировать пользовательские классы "Консольное приложение" и "Квадратная матрица", т.е. задать атрибуты и методы указанных классов, а также 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 «matrix».

В основном модуле консольного приложения (для языка С++ - 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 его методов должен выводить в консоль меню команд приложения, включающее:

- команду, инициирующую ввод с консоли значений, задающих объект матрицы (до ввода в программе должна быть задана матрица по умолчанию);

- команду, инициирующую расчет определителя матрицы и вывод результатов расчета;

- команду, инициирующую формирования транспонированной матрицы и ее вывода в консоль;

- команду, инициирующую расчет ранга матрицы и вывод результатов расчета;

- команду, инициирующую представление в консоль текущего объекта матрицы;

- команду выхода из приложения.

Модуль «matrix» должен содержать спецификацию класса "Квадратная матрица" и реализацию его методов, необходимых для достижения цели разрабатываемого приложения. Описание класса должно использовать вместо типа double (вещественное число, заданное в условии) абстрактный тип number, описание которого должно задаваться в отдельном заголовочном файле number.h с помощью оператора typedef double number (для С++).

ВАЖНО. Основное требование к реализации класса «matrix» заключается в том, что она должна быть инвариантна (одна и та же для различных вариантов использования) ко множеству применимых объектов. Здесь задано множество определения элементов матрицы как вещественное, а может быть и множества целых, комплексных, рациональных и т.д. Для различных множеств при сохранении функциональности реализация класса должна быть одна и та же. Если для различных множеств менять реализацию, то теряется смысл ООП как парадигмы.

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:**

*Класс приложения (TApplication)*

*Атрибуты*: нет

*Методы:*

* + - **exec** **-** обрабатывает выбор команды пользователем и

исполняет эту команду

*Типы параметров:* нет

*Тип возвращаемого значения:* int

*Область видимости:* public

* **menu** **-** выводит меню выбора команд

*Типы параметров:* нет

*Тип возвращаемого значения:* int

*Область видимости:* private

*Класс матрицы (TMatrix)*

*Атрибуты*:

* **numbers** **-** массив для хранения элементов матрицы

заданного пользователем размера

*Тип:* number\*\* (двумерный динамический массив типа

number)

*Область видимости:* private

* **n -** размер матрицы

*Тип:* int

*Область видимости:* private

*Методы:*

* + - **findDet -** вычисляет определитель матрицы

*Типы параметров:* нет

*Тип возвращаемого значения:* number

*Область видимости:* public

* **findRank -** вычисляет ранг матрицы

*Типы параметров:* нет

*Тип возвращаемого значения:* int

*Область видимости:* public

* + - **transpose -** транспонирует матрицу

*Типы параметров:* нет

*Тип возвращаемого значения:* void

*Область видимости:* public

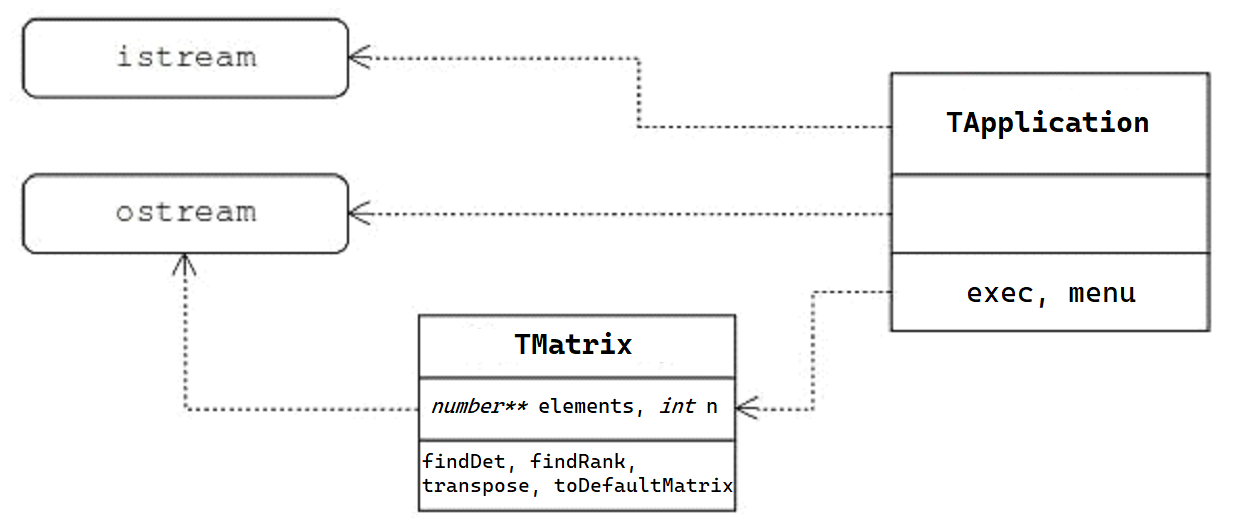
* **toDefaultMatrix** - приводит матрицу к виду по умолчанию (единичная матрица второго порядка)

*Типы параметров:* нет

*Тип возвращаемого значения:* void

*Область видимости:* public

**Диаграмма классов:**



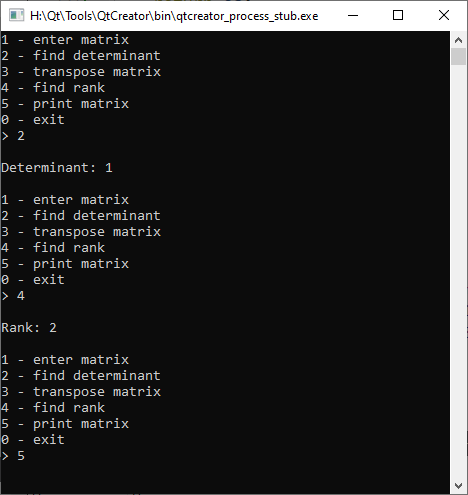
**Контрольные примеры:**

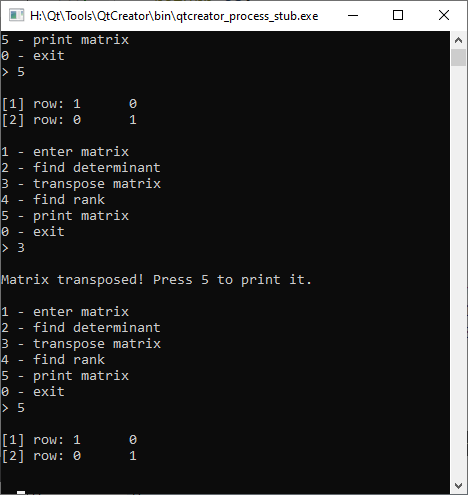
1. Данные по умолчанию (единичная матрица 2-го порядка)

*Ожидаемый результат:*

* "find determinant": вывод должен быть равен 1
* "find rank": вывод должен быть равен 2
* "print matrix": должна появиться единичная матрица 2-го порядка (вида [[1 0] [0 1]])
* "transpose matrix": при выводе матрицы ее вид не должен поменяться

*Полученный результат:*



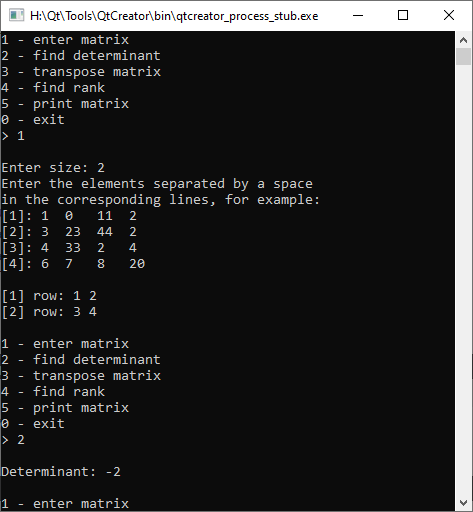


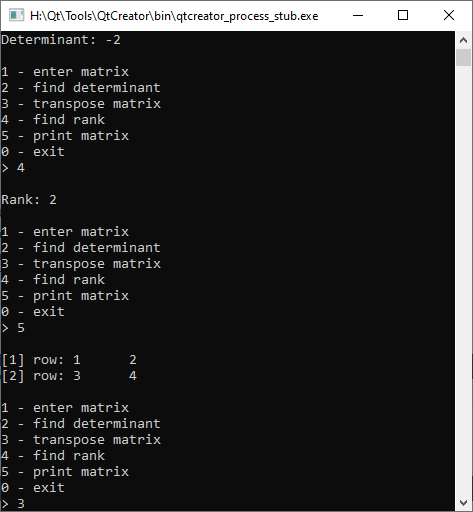
1. Ввод матрицы 2-го порядка вида [[1 2] [3 4]]

*Ожидаемый результат:*

* "find determinant": вывод должен быть равен -2
* "find rank": вывод должен быть равен 2
* "print matrix": должна появиться матрица 2-го порядка (вида [[1 2] [3 4]])
* "transpose matrix": при выводе матрицы ее вид должен поменяться на [[1 3] [2 4]]

*Полученный результат:*





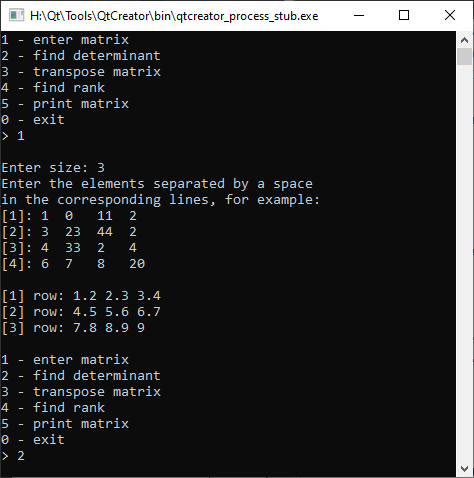


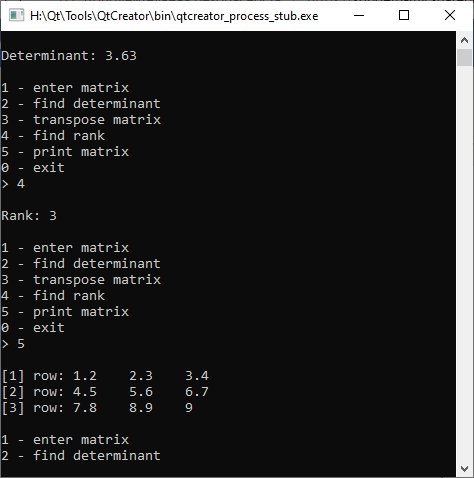
1. Ввод матрицы 3-го порядка вида [[1.2 2.3 3.4] [4.5 5.6 6.7] [7.8 8.9 9]]

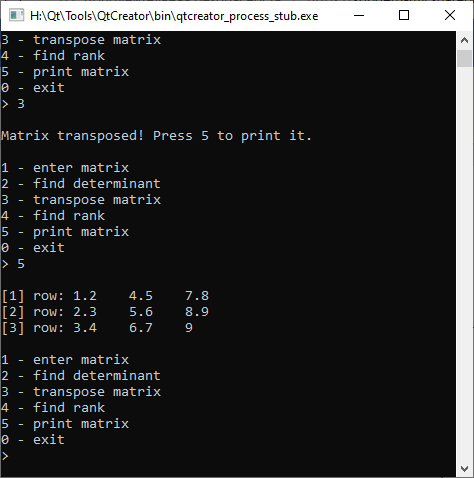
*Ожидаемый результат:*

* "find determinant": вывод должен быть равен 3.63
* "find rank": вывод должен быть равен 3
* "print matrix": должна появиться матрица 3-го порядка вида [[1.2 2.3 3.4] [4.5 5.6 6.7] [7.8 8.9 9]]
* "transpose matrix": при выводе матрицы ее вид должен поменяться на [[1.2 4.5 7.8] [2.3 5.6 8.9] [3.4 6.7 9]]

*Полученный результат:*





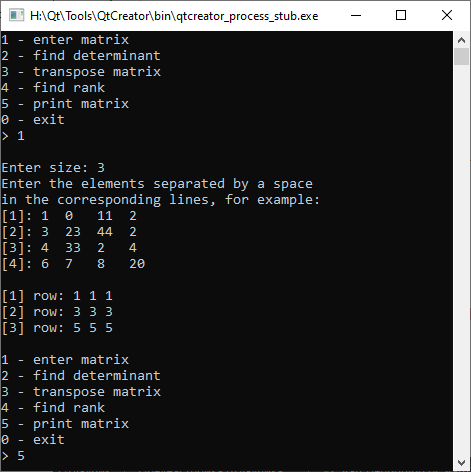


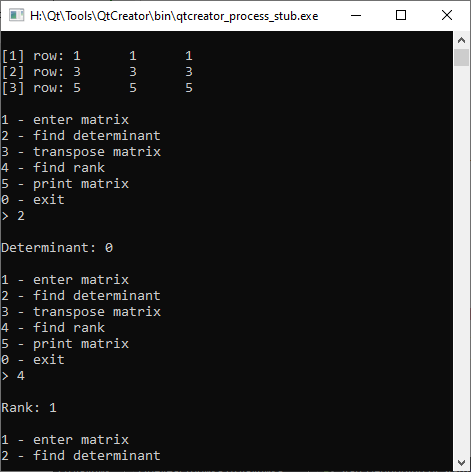
1. Ввод матрицы 3-го порядка вида [[1 1 1] [3 3 3] [5 5 5]]

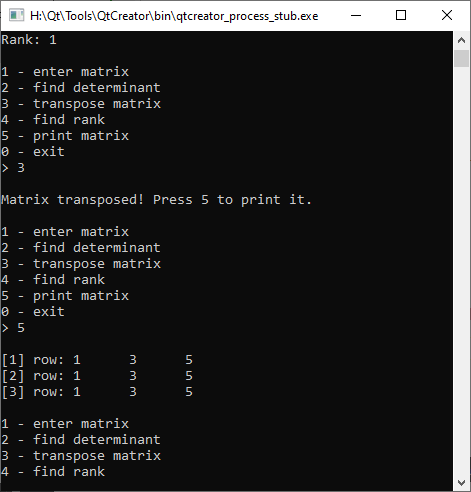
*Ожидаемый результат:*

* "find determinant": вывод должен быть равен 0
* "find rank": вывод должен быть равен 1
* "print matrix": должна появиться матрица 3-го порядка вида [[1 1 1] [3 3 3] [5 5 5]]
* "transpose matrix": при выводе матрицы ее вид должен поменяться на [[1 3 5] [1 3 5] [1 3 5]]

*Полученный результат:*







**Выводы:**

В ходе выполнения практической работы было реализовано приложение для выполнения базовых операций над матрицами (вычисление определителя и ранга, транспонирование, вывод матрицы в консоль) с помощью классов приложения и матрицы. Приложение было протестировано на контрольных примерах, в ходе которых была подтверждена корректность его работы.