

Санкт-Петербургский Государственный Электротехнический Университет
«ЛЭТИ» им. В. И. Ульянова (Ленина)

Кафедра информационных систем

Отчет

По практической работе №2

По дисциплине «Объектно-ориентированное программирование»

Студенты группы 2372 _____ Тубшинов В. Т., Алексеев Г.

Преподаватель _____ Егоров С. С.

г. Санкт-Петербург

2023 г.

Задание: Создать консольное приложение согласно представленной на рис.1 диаграмме классов, предназначенное для заданных вычислений над квадратной матрицей, заданной на множестве комплексных чисел.

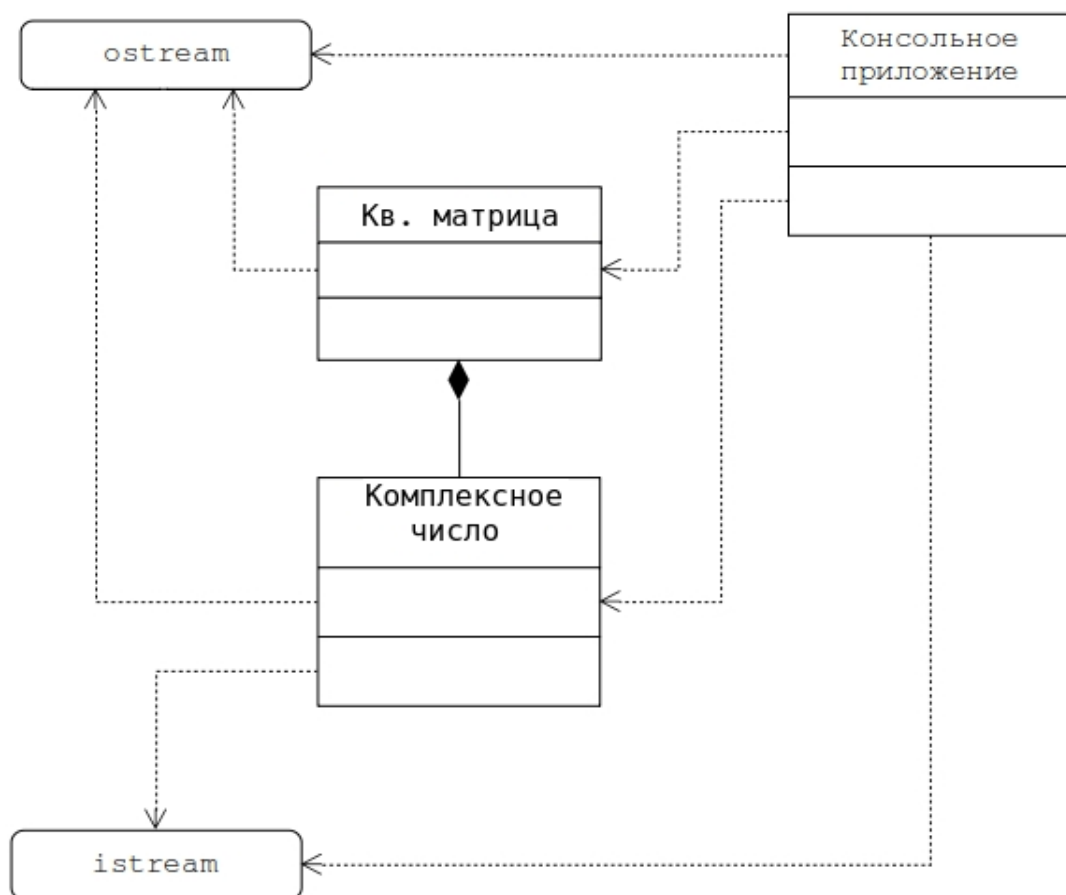


рис. 1

Специфицировать пользовательские классы «Консольное приложение», «Квадратная матрица» и «Комплексное число», задать атрибуты и методы указанных классов, а также распределить их по существующим областям видимости.

Приложение должно включать основной модуль (функция main), модуль «application» и модуль «matrix», а также модуль «complex».

В основном модуле консольного приложения 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, иницирующую ввод с консоли значений, задающих объект матрицы (до ввода в программе должна быть задана матрица по умолчанию);
- команду, иницирующую расчет определителя матрицы и вывод результатов расчета;
- команду, иницирующую формирования транспонированной матрицы и ее вывода в консоль;
- команду, иницирующую расчет ранга матрицы и вывод результатов расчета;
- команду, иницирующую представление в консоль текущего объекта матрицы;
- команду выхода из приложения.

Спецификации классов.

Class Application:

Методы:

- `int menu() (private)` – функция печатает в консоль то, что пользователю доступно для выбора.
- `Application() (public)` – конструктор класса
- `int exec() (public)` – функция, имитирующая консоль, где происходят все команды.

Class Matrix:

Методы:

- `number determinant(number matrix[3][3]) (public)` – вызывается в конструкторе при создании объекта и ищет определитель матрицы через миноры (рекурсивный алгоритм, который раскладывает все

миноры по первой строчке). В результате отдает вещественное число – определитель матрицы;

- `matrix()` (public) - конструктор класса;
- `int rankOfMatrix3x3(number matrix[3][3])` (public) – вызывается в конструкторе при создании объекта и ищет ранг матрицы через миноры (ищет наибольший минор, определитель которого не равен нулю). В результате отдает целое число – ранг матрицы;
- `void transposition(number matrix[3][3])` (public) – метод транспонирует матрицу;
- `void show(number matrix[3][3])`(public) – функция для вывода матрицы на экрана;
- `void input(number matrix[3][3])`(public) - функция для ввода значений в матрицу.

class Complex:

Атрибуты:

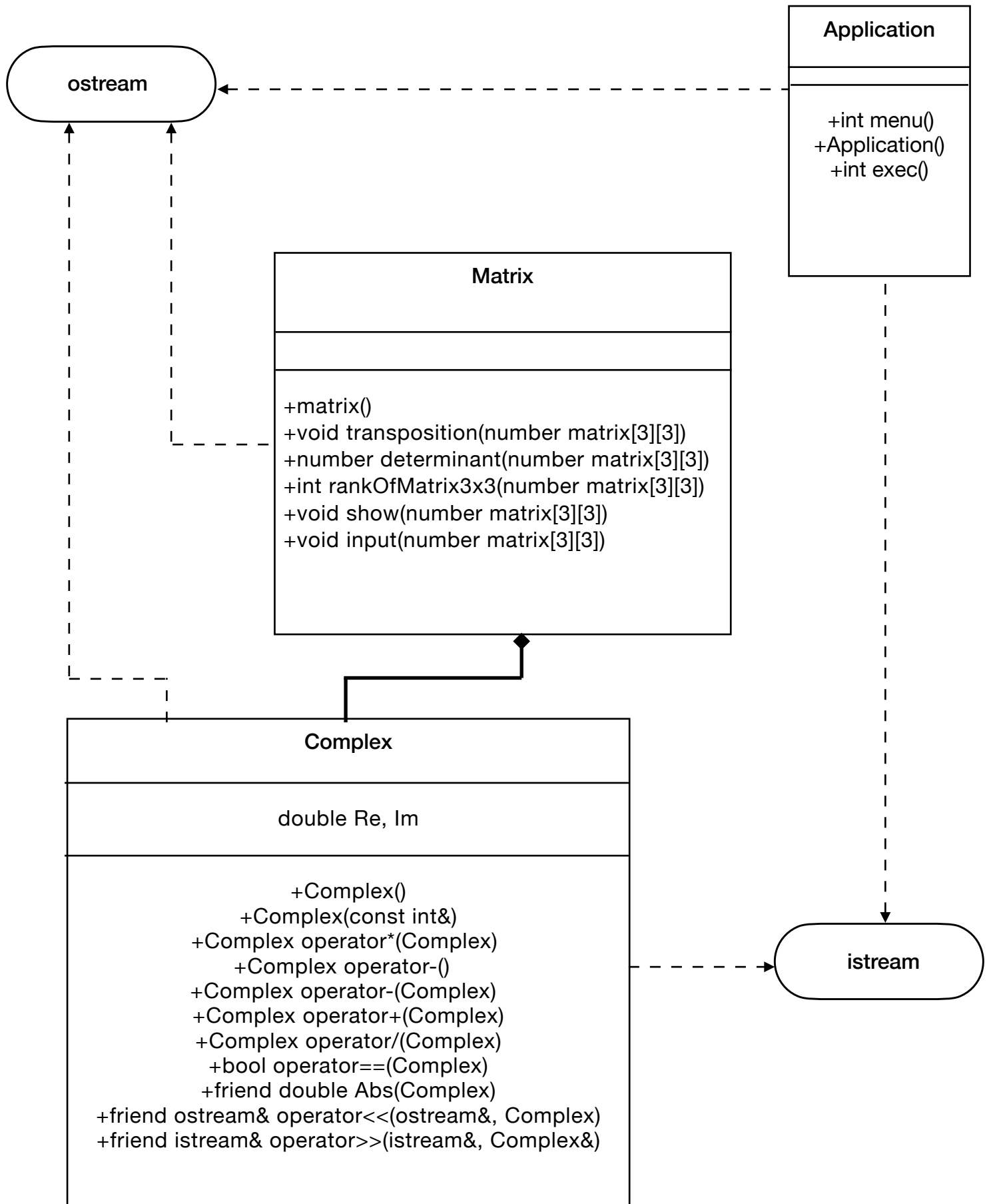
Double Im - мнимая часть числа

Double Re - вещественная часть числа

Методы:

- `Complex()`(public) - конструктор класса
- `Complex (const int&)` - конструктор комплексного числа(объекта)
- `Complex operator*(Complex)` - функция для умножения комплексных чисел
- `Complex operator-(Complex)` - функция вычитания комплексных чисел

- `Complex operator-()` - унарная операция равная умножению на -1
- `Complex operator/(Complex)` - функция деления комплексных чисел
- `Complex operator+(Complex)` - функция сложения комплексных чисел
- `bool operator==(Complex)` - функция равенства(возвращает true, если два числа равны, и false, если нет)
- `friend Abs(Complex)` - функция для нахождения модуля комплексного числа
- `friend ostream& operator<<(ostream&, Complex)` - функция вывода комплексного числа
- `friend istream& operator>>(istream&, Complex&)` - функция ввода комплексного числа



Контрольные примеры:

Исходная матрица:

$3-4i$	1	$4-8i$
$6+i$	$5+3i$	9
$11+2i$	$6-5i$	$-8+3i$

Определитель:

$$\text{Det}(A) = -583 + 324i$$

Ранг:

$$\text{Rank}(A) = 3$$

Транспонирование:

$$\mathbf{A}^T = \begin{array}{|c|c|c|} \hline 3-4i & 6+i & 11+2i \\ \hline 1 & 5+3i & 6-5i \\ \hline 4-8i & 9 & -8+3i \\ \hline \end{array}$$

Работа матрицы на контрольных примерах:

```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 1
  Re Im
1)3 -4
2)1 0
3)4 -8
4)6 1
5)5 3
6)9 0
7)11 2
8)6 -5
9)-8 3
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 5
The matrix:
  3-4i   1+0i   4-8i
  6+1i   5+3i   9+0i
  11+2i   6-5i  -8+3i
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 2
The value of determinant is -583+324i
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 5
The matrix:
  3-4i   1+0i   4-8i
  6+1i   5+3i   9+0i
  11+2i   6-5i  -8+3i
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 3
The matrix is transposed
1 - input new values(coefficients)
2 - value of the determinant
3 - transposed matrix
4 - the rank of the matrix
5 - output values of the matrix
0 - exit
> 5
The matrix:
```

Активация Windows
Чтобы активировать Windows, перейдите в раздел "Параметры".

