

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И ТЕЛЕКОММУНИКАЦИЙ
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ

Лабораторная работа №2
по дисциплине
«Информационные технологии и программирования»

Выполнил:
Переверза Владислав Александрович
Студент 1 курса группы ПИН-б-о-22-1
Направления подготовки
09.03.03 Прикладная информатика
очной формы обучения

Ставрополь, 2023 г.

Тема: Классы

Цель работы: изучить базовые понятия (классы, подклассы и методы) Реализовать фундаментальные принципы объектно-ориентированного программирования.

Ход работы

Вариант - 17

Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами:

- сложение, вычитание, умножение, деление (+, −, *, /) (умножение и деление как на другую матрицу, так и на число);
- комбинированные операции присваивания (+=, -=, *=, /=);
- операции сравнения на равенство (неравенство);
- операции вычисления обратной и транспонированной матрицы,
- операцию возведения в степень;
- методы вычисления детерминанта и нормы;
- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметрическая, верхняя треугольная, нижняя треугольная);
- операции ввода-вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

Листинг приведён в файлах:

[main.cpp](#)

[Matrix.h](#)

[Matrix.cpp](#)

[MatrixTest.h](#)

[MatrixTest.cpp](#)

Также приведена [UML-диаграмма](#) проекта

В заголовочном файле *Matrix.h* определён класс *Matrix*, в файле *Matrix.cpp* приведена реализация класса *Matrix*. В заголовочном файле *MatrixTest.h* определены функции тестирования, в файле *MatrixTest.cpp* приведены реализации функций тестирования. В файле *main.cpp* реализована функция *main*, предоставляющая интерфейс к классу и проводящая тестирование кода.

Ссылка на [репозиторий](#), содержащий полностью выполненные задания.

Вывод: изучил основы объектно-ориентированного программирования, в том числе понятия классов, подклассов и методов. Также реализовал основные принципы этого подхода на практике.