# Введение

Настоящая техническое задание определяют требования к программному продукту калькулятора матрицы «BadMatrixCalculator». Программа предназначена для выполнения основных операций линейной алгебры над матрицами, таких как сложение, вычитание, умножение на число и на матрицу, транспортирование, а также вспомогательных действий по вводу и отображения данных.

# Основания для разработки

Разработка программы «BadMatrixCalculator» выполняется в рамках учебного проекта по предмету “Тестирование и верификация программного обеспечения”. Необходимость создания продукта обусловлена учебным заданим по дисциплине «Тестирование и верификация программного обеспечению».

# Назначение разработки

Программа «BadMatrixCalculator» предназначена для выполнением математических операций над матрицами. Основная задача разработки заключается в обеспечении пользователя удобным инструментом для выполнения базовых операции над матрицами (сложение, вычитание, умножение, транспортирование, умножение на скаляр).

# Требование к программе

1. Функциональные требования

Программа «BadMatrixCalculator» должна обеспечивать:

* Ввод матрицы с клавиатуры пользователем;
* Отображение введенной матрицы на экране;
* вычисления отрицательной (противоположной) матрицы;
* умножение матрицы на скаляр;
* транспонирование матрицы;
* сложение двух матриц;
* Вычитание одной матрицы из другой;
* умножение двух матриц;
* вывод результатов вычисления в консоль в удобном для чтения формате;

1. Требования к надежности

* Программа должна корректно завершать работу при некорректном вводе данных, исключая аварийное завершение.
* При возникновении ошибки ввода (Например попытка вести символ вместо числа) программа должна запрашивать повторный ввод.
* В случае несоответствия размеров матрицы для выполнения операции должно выводиться сообщение об ошибке.

1. Условия эксплуатации

* Программа предназначена для эксплуатации в учебных и демонстрационных целях.
* Среда эксплуатации: персональный компьютер.
* Требуемая версия среды выполнения JAVA SE 8 или выше.
* Минимальная аппаратные требования: 2 ГБ оперативной памяти, процессор с тактовой частотой не менее 1,5 ГГц.

1. Совместимость

* Программа должна работать на операционных системах Windows, Linux, macOS, при наличии установленной Java Virtual Machine (JVM).
* Взаимодействие с другими программными системами не требуется.

# Требования к интерфейсу

Интерфейс программы «BadMatrixCalculator» реализуется в виде консольного меню.

* После запуска программы на экране должно отображаться текстовое меню с перечнем доступных операций:
  1. Ввод матрицы
  2. отображение матрицы
  3. нахождение противоположной матрицы
  4. умножение на скаляр
  5. транспортирование матрицы
  6. сложение матриц
  7. вычитание матриц
  8. умножение матриц
  9. выход из программы
* Пользователь выбирает действия вводом соответствующего номера.
* Для операций, требующих дополнительных данных система должна запрашивать их через консоль (например размеры матрицы, элементы матрицы, значение скаляра).
* При некорректном вводе пользователь получает сообщение об ошибке и повторный запрос ввода.
* Результаты выполнения операции отображаются в текстовом виде.

# Критерии приемки

Программа «BadMatrixCalculator» требует выполнения следующих условий:

* Все заявленные функции (ввод, отображение, отрицание, умножение на скаляр, транспонирование, сложение, вычитание, умножения матриц) корректно работают при правильном вводе данных.
* Программа корректно обрабатывает неправильный ввод (символы вместо чисел, несоответствие размеров матриц) с выдачей сообщений об ошибке без аварийного завершения.
* Успешное выполнение не менее 95% тестовых сценариев, разработанных на основе функциональных требований.
* Программа запускается и выполняется на OC Windows, Linux и macOS При наличии установленной JVM версии 8 или выше.
* Время отклика программы при вводе и выводе данных не превышает одну секунду для матриц размером до 20х20.

# Требование к документации

Для программы «BadMatrixCalculator» должны быть подготовлены следующие документы

1. Руководство пользователя

* Описание назначения программы;
* Пошаговая инструкция по запуску использованию всех функций;
* Примеры ввода данных и получения результатов;
* Раздел «Обработка ошибок» с использованием возможных сообщений и действий пользователя.

2. Описание архитектуры системы

# Порядок контроля и приемки

Контроль и приемка программы «BadMatrixCalculator» осуществляется следующими методами:

1. Функциональное тестирование – метод чёрного ящика
2. Структурное тестирование – метод белого ящика, метод сер
3. Тестирование совместимости

* Проверка работы программы на OC Windows, Linux и macOS при установленной JVM версии 8 и выше.

1. Приемочные испытания

* Составление тестовых сценариев с разными размерами матриц (например 2х2, 3х3, 5х5) и различными операциями.
* Сравнение фактических результатов с ожидаемыми.
* Подтверждение соответствия программы требованиям по времени отклика (не более 1 секунды для матриц до 20х20).

1. Критерии успешной приемки

* Все функциональные тесты выполнены корректно;
* Обработка ошибок осуществляется без аварийного завершения;
* Программа соответствует требованиям совместимости и эксплуатационным условиям;
* Успешное выполнение 95% тестовых сценариев.

# Этапы и сроки разработки

Разработка программы «BadMatrixCalculator» Выполняется в течение одной недели с 02.09.25 по 09.09.25. План-график реализации проекта включает следующие этапы:

1. 2.09.25 - Анализ требований и составления технического задания

* Определение функциональных возможностей программы
* Разработка структуры классов интерфейса

1. 3-5.09.25 - Написание исходного кода программы

* Реализация классов и методов для операций с матрицами

1. 6-7.09.25 - Тестирование и отладка программы

* Проверка корректного выполнения всех операций с матрицами
* обработка ошибок при некорректном воде
* проверка работы с различными ОС

1. Подготовка пользовательской и технической документации

* Руководство пользователя;
* Техническое описание;
* Тестовая документация с примерами

1. 09.09.25 - приемка и сдача программы заказчику

* Проведение приемочных испытаний;
* Подтверждение соответствия функциональным требованиям;
* Финальная задача программы.