**Класс Matrix. Проектная документация.**

**32/2 группа.**

1. Назначение класса.

Класс Matrix предназначен для работы с матрицами произвольных размеров, квадратных и прямоугольных.

Благодаря классу пользователь может использовать матрицы как це­лост­ные объекты, без необходимости изучения реализации предусмотренных классом операций.

1. Набор реализуемых операций

Создание и уничтожение объектов-матриц.

Операция присваивания матриц. (A = B)

Сложение, вычитание и перемножение матриц. (A + B; A – B;   
A \* B;).

Прибавление, убавление и домножение матриц. (A += B; A -= B;   
 A \*= B;)

Транспонирование матриц. (Transpose)

Проверка матриц на диагональность. (Diag)

Реализация трех элементарных преобразований матрицы. (SwapRows; ProdRow; SumRows)

Вставка и извлечение матрицы из другой матрицы. (Insert;   
Extract)

Доступ к элементу матрицы по индексу.

Нахождение определителя матрицы. (Det)

1. Структура данных объекта

Объект должен иметь и (…). Члены матрицы будем размещать в динамической памяти построчно от до ; от до … от до , используя одномерный массив. Обращение к нему производится при помощи указателей.

Таким образом объект должен иметь 4 поля:

**unsigned** **int** rows, cols; – количество строк (rows) и столбцов (cols);

**double** \* elem; – если под массив выделена динамическая память, указатель содержит ссылку начальный элемент памяти. Если же динамическая память не выделена, то rows = 0, cols = 0, elem = NULL;

**bool** simpleCopy;.

Эти поля являются **protected** и не доступны пользователю.

1. Конструкторы

Пустой конструктор:

Matrix();

Не содержит в себе никакой информации и не выделяет динамическую память. Нужен для объявления типа переменной.

Конструктор квадратных матриц:

Matrix(**unsigned** n, **double** \*p);

Конструктор прямоугольных матриц:

Matrix(**unsigned** n,**unsigned** m, **double** \*p);

Копирующий конструктор, который создает объект, являющийся копией объекта:

Matrix(**const** Matrix& A);

Считывающий с файла конструктор:

Matrix(FILE \* fp);

Считывает матрицу в виде одномерного массива из файла. Тип файла .txt. Файл должен содержать

Также есть секретные конструкторы, которые не доступны для пользователя.

Все конструкторы возвращают значение simpleCopy = false.

1. Деструктор

Предназначен для освобождения динамической памяти, после работы деструктора.

Matrix::~Matrix();

1. Операторы

Matrix& **operator**= (**const** Matrix& A);

Оператор присваивания. Компилятор будет превращать A = B; в вызов метода A.**operator**=(B);

**double**& **operator**[](**unsigned** i) **const**;

Оператор квадратные скобки. Так как мы работаем с двумерными матрицами, то доступ к коэффициентам матрицы производится с использованием двух параметров: a = A[i][j];

1. Метод отладки

**void** Display() **const**; - служит для вывода двумерной матрицы в консоль.

1. Операции
   1. Сложение матриц
   2. Вычитание матриц
   3. Прибавление
   4. Домножение
   5. Умножение матрицы на константу
   6. Перемножение матриц
   7. Проверка матрицы на диагональность
   8. Транспонирование матриц
   9. Перестановка местами любых двух строк матрицы
   10. Умножение строки матрицы на не нулевую константу
   11. Прибавление к любой строке матрицы другой строки, умноженной на не нулевую константу
   12. Вставка матрицы в другую матрицу
   13. Извлечение матрицы из другой матрицы
   14. Нахождение определителя матрицы