**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по домашнему заданию

«Метод расчёта определителя матрицы F#»

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| Студенты группы ИУ5-32Б: |  | преподаватель каф. ИУ5 |
| Бекетов Роман Александрович  Семчёнок Владимир Дмитриевич |  | Гапанюк Ю.Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Задание**

Разработать программу для подсчёта определителя матрицы методом Гаусса. Замер производительности.

**Листинг**

let stopWatch = System.Diagnostics.Stopwatch.StartNew()

type Matrix = { values: float[,] }

with

static member ofArray2D (values: float [,]) =

{ values = values }

end

type result =

| Error of string

| SuccessZero of int

| None

let rec checkZero(A:Matrix, rows:int, cols:int, size:int):int =

if (rows >= size) then

-1

else

if (A.values[rows, cols] = 0.) then

checkZero(A, rows+1, cols, size)

else

rows

let rec swapRows (A:Matrix, firstRows:int, secondRows:int, size:int, cols:int):result =

if (firstRows >= size secondRows >= size cols >= size) then

None

else

let temp = A.values[firstRows, cols]

A.values[firstRows, cols] <- A.values[secondRows, cols]

A.values[secondRows, cols] <- temp

swapRows(A, firstRows, secondRows, size, cols+1)

let rec subCols(A:Matrix, rows:int, cols:int, size:int, koef:float, origRows:int):result =

if (cols = size) then

None

else

A.values[rows, cols] <- (A.values[rows, cols] - koef\*A.values[origRows, cols])

subCols(A, rows, cols+1, size, koef, origRows)

let rec subRows(A:Matrix, rows:int, cols:int, size:int, origRows:int, origCols:int):result =

if (rows = size) then

None

else

let koef = A.values[rows, cols] / A.values[origRows, origCols]

subCols(A, rows, cols, size, koef, origRows) |> ignore

subRows(A, rows+1, cols, size, origRows, origCols)

let rec Umatrix (A:Matrix, rows:int, cols:int, size:int):int =

if (rows >= size) then

1

else

if (A.values[rows, cols] = 0.) then

let ZeroResult = checkZero(A, rows+1, cols, size)

if (ZeroResult = -1) then

Error ("in func Zero: rows bigger size")

else

swapRows(A,rows,ZeroResult,size,0)

subRows(A, rows+1, cols, size, rows, cols) |> ignore

else

subRows(A, rows+1, cols, size, rows, cols) |> ignore

Umatrix(A, rows+1, cols+1, size)

let rec det(A:Matrix, rows:int, cols:int, size:int, det\_):float =

if (rows = size) then

det\_

else

det(A, rows+1, cols+1, size, det\_\*A.values[rows, cols])

// main

let size = 4

let a = array2D [[12.;4.;2.; 1.]

[4.;1.;6.; 34.]

[10.;7.;4.; 5.]

[3.;5.;4.; 3.]]

let A = Matrix.ofArray2D a

printfn "Matrix before"

printfn "%A\n" A

Umatrix(A, 0, 0, size)

let detetm = det (A, 0, 0, size, 1);

printfn "Matrix after"

printfn "%A\n" A

printfn "Determinant of matrix"

printfn "%A" detetm

printfn "Measure time:"

stopWatch.Stop()

printfn "%f" stopWatch.Elapsed.TotalMilliseconds

**Тесты**

****