

**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:  
Студенты группы ИУ5-32Б:  
Бекетов Роман Александрович  
Семчёнок Владимир Дмитриевич  
Подпись и дата:

Проверил:  
преподаватель каф. ИУ5  
Гапанюк Ю.Е.

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

```
13.000000
• vladimir@MiPro--VAVASTO:~/Desktop/f_det$ dotnet fsi det.fsx

/home/vladimir/Desktop/f_det/det.fsx(52,13): warning FS0020: The result of this expression has type 'result' and is implicitly ignored. Consider using 'ignore' to discard this value explicitly, e.g. 'expr |> ignore', or 'let' to bind the result to a name, e.g. 'let result = expr'.

Matrix before
{ values = [[12.0; 4.0; 2.0; 1.0]
            [4.0; 1.0; 6.0; 34.0]
            [10.0; 7.0; 4.0; 5.0]
            [3.0; 5.0; 4.0; 3.0]] }

Matrix after
{ values = [[12.0; 4.0; 2.0; 1.0]
            [0.0; -0.3333333333; 5.333333333; 33.66666667]
            [0.0; 0.0; 61.0; 374.5]
            [0.0; 0.0; 0.0; -7.655737705]] }

Determinant of matrix
1868.0
Measure time:
14.073700
```