

**Лабораторная работа № 15**  
**Выполнение матричных операций. Программа аутентификации**  
**пользователя.**

**Задание 1. Написание библиотеки матричных операций**

При решении задач моделирования экономических и производственных процессов во многих случаях встречаются вычисления связанные с матричными операциями. Требуется написать программу, выполняющую следующую математическую операцию (согласно варианту задания):

- 1) нахождение минора
- 2) нахождение определителя
- 3) нахождение обратной матрицы
- 4) нахождение произведения вектора на вектор
- 5) нахождение произведения матрицы на вектор
- 6) нахождение суммы матриц одинаковой размерности
- 7) нахождение суммы векторов одинаковой размерности

Ниже в качестве примера приведена возможная программная реализация решения перечисленных выше задач. Отметим, что для простоты реализации размерность участвующих в выполнении операции операндов передается в процедуры и функции в качестве параметров, а сами матрицы и векторы, как входные, так и результирующие передаются по ссылке.

```
Sub sum_matr(ByRef a() As Double, ByRef b() As Double,  
ByRef mo() As Double, x As Integer, y As Integer)
```

```
'суммирование матриц a и b, результат в c (x строк y столбцов)
```

```
For i = 0 To x - 1
```

```
For j = 0 To y - 1
```

```
mo(i) = a(i) + b(i)
```

```
Next j
```

```
Next i
```

```
End Sub
```

```
Sub sum_vect(ByRef vi1() As Double, ByRef vi2() As  
Double, ByRef vo() As Double, x As Integer)
```

```
'суммирование векторов vi1 и vi2, результат в vo (x разрядов)
```

```
For i = 0 To x - 1
```

```
vo(i) = vi1(i) + vi2(i)
```

```
Next i
```

```
End Sub
```

```
Sub matr_x_vect(ByRef a() As Double, ByRef vi() As  
Double, ByRef vo() As Double, x As Integer, y As In-  
teger)
```

'произведение матрицы a на вектор vi, результат в vo ' (размерность матрицы: x строк, y столбцов, вектора: y разрядов)

```
For i = 0 To x - 1
vo(i) = 0
For j = 0 To y - 1
vo(i) = vo(i) + a(i, j) * vi(j)
Next j
Next i
End Sub
```

```
Function vect_x_vect(ByRef vi1 As Double, vi2 As Double, z As Integer) As Double
```

' функция возвращает результат скалярного произведения векторов vi1 и vi2, размерностью z разрядов

```
vect_x_vect = 0
For i = 0 To z - 1
vect_x_vect = vect_x_vect + vi1(i) * vi2(i)
Next i
End Function
```

```
Function minor2(ByRef a() As Double) As Double
```

' функция вычисляет определитель 2x2

```
minor2 = a(0, 0) * a(1, 1) - a(0, 1) * a(1, 0)
End Function
```

```
Sub tominor(ByRef a() As Double, ByRef b() As Double, ii As Integer, jj As Integer, zz As Integer)
```

'процедура получает минор для ii, jj элемента матрицы a и записывает в матрицу b

```
For i = 0 To zz
For j = 0 To zz
If i < ii And j < jj Then
b(i, j) = a(i, j)
Else
If i < ii And j > jj Then
```

```
b(i, j - 1) = a(i, j)
```

```
Else
```

```
If i > ii And j < jj Then
```

```
b(i - 1, j) = a(i, j)
```

```
Else
```

```
If i > ii And j > jj Then
```

```
b(i - 1, j - 1) = a(i, j)
```

```
End If
```

```
End If
```

```
End If
```

```
End If
```

```
Next j
```

```

Next i
End Sub
Sub show_matr(ByRef a() As Double, x As Integer, y As Integer, Optional st As String)
'отображает матрицу a размерности x, y в окне MsgBox
Dim ss As String
For i = 0 To x - 1
For j = 0 To y - 1
ss = ss & Round(a(i, j), 3) & vbTab
Next j
ss = ss & vbLf
Next i
MsgBox ss, , st
End Sub
Sub show_vect(ByRef v() As Double, x As Integer, Optional st As String)
'отображает вектор v (x разрядов)
Dim ss As String
For i = 0 To x - 1
ss = ss & Round(v(i), 3) & vbTab
Next i
MsgBox ss, , st
End Sub
Function minorx(ByRef a() As Double, ii As Integer) As Double
'вычисляет определитель квадратной матрицы a размерности ii x ii

Dim m(10, 10) As Double
Dim i As Integer
Dim ss As Integer
Dim mm As Integer
ss = 0
For i = 0 To ii - 1
Call tominor(a, m, 0, i, ii - 1)
If ii = 3 Then
mm = minor2(m) * (-1) ^ (i Mod 2)
Else
mm = minorx(m, ii - 1) * (-1) ^ (i Mod 2)
End If
ss = ss + a(0, i) * mm
Next i
minorx = ss
End Function
Sub transp_matr(ByRef a() As Double, ByRef b() As Double, xx As Integer)
'транспонирует квадратную матрицу a, результат в b (размерность xx)

```

```

For i = 0 To xx - 1
For j = 0 To xx - 1
b(i, j) = a(j, i)
Next j
Next i
End Sub
Sub inv_matr(ByRef a() As Double, ByRef b() As Double,
xx As Integer)
'находит матрицу b обратную квадратной матрице a размерности xx
Dim c(10, 10) As Double
Dim bb(10, 10) As Double
Dim i As Integer, j As Integer
Dim det_x As Double
det_x = minorx(a, xx)
If det_x = 0 Then
MsgBox "det=0"
Exit Sub
End If
For i = 0 To xx - 1
For j = 0 To xx - 1

Call tominor(a, c, i, j, xx - 1)
If xx = 3 Then
bb(i, j) = minor2(c) * (-1) ^ (i + j)
Else
bb(i, j) = minorx(c, xx - 1) * (-1) ^ (i + j)
End If
Next j
Next i
Call transp_matr(bb, b, xx)
For i = 0 To xx - 1
For j = 0 To xx - 1
b(i, j) = b(i, j) / det_x
Next j
Next i
End Sub
Sub mmz( )
'процедура, показывающее «создание» и решение систем линейных
уравнений
Dim i As Integer
Dim a() As Variant
Dim b() As Variant
Dim z() As Variant
Dim c(3, 3) As Double
Dim d(4, 4) As Double
Dim e(5, 5) As Double

```

```

Dim bb(10, 10) As Double
Dim vz(10) As Double
Dim vv3 As Variant
Dim vx3(3) As Double
Dim vb3(3) As Double
Dim vv4 As Variant
Dim vx4(4) As Double
Dim vb4(4) As Double
Dim vv5 As Variant
Dim vx5(5) As Double
Dim vb5(5) As Double

vv3 = [{1,2,3}] ' инициализация вектора размерности 3
vv4 = [{1,2,3,4}] ' инициализация вектора размерности 4
vv5 = [{1,2,3,4,5}] ' инициализация вектора размерности 5
a = [{1,2,3;4,1,6;7,8,9}] ' инициализация матрицы 3x3
b = [{1,2,3,4;4,1,6,7;7,8,9,10;11,11,12,13}] '
инициализация матрицы 4x4
z = [{1,2,3,4,5;4,1,6,7,8;7,8,9,10,11;11,11,12,13,14;
15,16,17,18,18}]
' инициализация матрицы 5x5
' При инициализации нумерация элементов начинается с 1, а нам надо с 0
' Новыми матрицами будут матрицы c, d и e
For i = 0 To 2
  vx3(i) = vv3(i + 1)
  For j = 0 To 2
    c(i, j) = a(i + 1, j + 1)
  Next j
Next i
For i = 0 To 3
  vx4(i) = vv4(i + 1)
  For j = 0 To 3
    d(i, j) = b(i + 1, j + 1)
  Next j
Next i
For i = 0 To 4
  vx5(i) = vv5(i + 1)
  For j = 0 To 4
    e(i, j) = z(i + 1, j + 1)
  Next j
Next i
'Находим векторы vb для составления линейных уравнений A*x=vb
Call matr_x_vect(c, vx3, vb3, 3, 3)
Call show_vect(vb3, 3, "Вектор b3")

```

```

Call matr_x_vect(d, vx4, vb4, 4, 4)
Call show_vect(vb4, 4, "Вектор b4")
Call matr_x_vect(e, vx5, vb5, 5, 5)

Call show_vect(vb5, 5, "Вектор b5")
'Находим обратную матрицу для матрицы c
Call inv_matr(c, bb, 3)
Call show_matr(bb, 3, 3, "Матрица обратная c")
'Находим решение уравнения
Call matr_x_vect(bb, vb3, vz, 3, 3)
Call show_vect(vz, 3, "Решение уравнения")
'Находим обратную матрицу для матрицы d
Call inv_matr(d, bb, 4)
Call show_matr(bb, 4, 4, "Матрица обратная d")
'Находим решение уравнения
Call matr_x_vect(bb, vb4, vz, 4, 4)
Call show_vect(vz, 4, "Решение уравнения")
'Находим обратную матрицу для матрицы e
Call inv_matr(e, bb, 5)
Call show_matr(bb, 5, 5, "Матрица обратная e")
'Находим решение уравнения
Call matr_x_vect(bb, vb5, vz, 5, 5)
Call show_vect(vz, 5, "Решение уравнения")
End Sub

```

## **Задание 2. Написание программы аутентификации пользователя**

Программная реализация задачи аутентификации пользователя предполагает решение следующих подзадач:

- 1) задание пользовательского пароля;
- 2) сохранение пароля в защищенном виде;
- 3) проверка на соответствие логина и пароля.

В свою очередь, одно из решений подзадачи сохранения пароля в защищенном виде предполагает преобразование пароля при помощи односторонней функции, которую мы назвали квазихэш функцией из-за ее простоты и неполного соответствия требованиям, предъявляемым к реальным хэшфункциям. В качестве односторонней функции мы использовали операцию нахождения остатка по модулю какого-то простого числа  $a$  от 30 до 110 (операция  $\text{mod } a$ ) для кода каждого из символов пароля. Форма для выполнения описанных выше операций может выглядеть таким образом, как это показано на рисунке Рис.1.

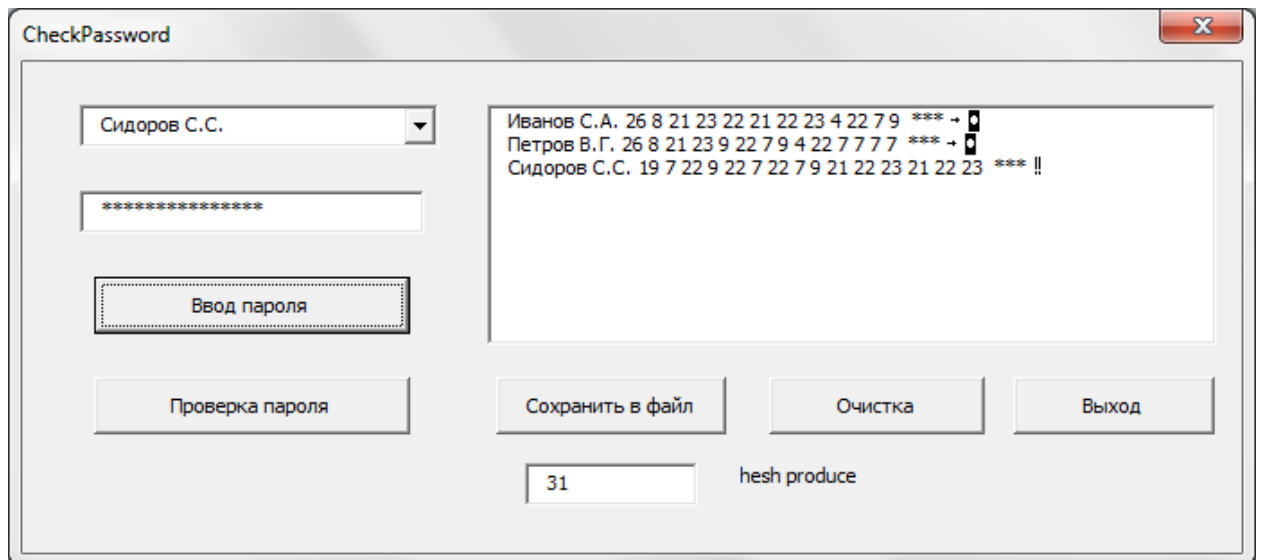


Рис.1

В качестве пользователей рассмотрены для наглядности только три пользователя: Иванов, Петров и Сидоров, которые заносятся в элемент управления ComboBox во время инициализации формы. При реализации программы необходимо указать фамилию и инициалы исполнителя и еще двух студентов группы. Для наглядности после очередного изменения пароля логин и хэш пароля должны отображаться в элементе управления ListBox. Для упрощения выполнения файловых операций чтения /записи запись хэшей паролей в файл разрешена только после того, как будут заполнены все пароли.

Операция проверки соответствия логина и пароля проводится в несколько этапов. Файл с паролями читается построчно и производится поиск подстроки содержащей идентификатор выбранный в ComboBox. При совпадении идентификаторов выполняется проверка длины проверяемого пароля и длины хэш для данного логина, записанного в файл. При совпадении и этих параметров проводится посимвольное сравнение. Для придания реалистичности разрабатываемому проекту можно поставить дополнительную задачу ведения протоколирования в текстовый файл всех попыток (и удачных и не удачны) проверки пароля.

Ниже показан пример кода программы, реализующей данный проект. Отметим, что вся программа реализована в виде функций пользовательской формы.

```
Dim pw(3, 20) As Integer
Dim Names(3) As String
Dim try_pwd(200) As String
Dim try_pwd_ind As String
Dim hesh_x As Integer
Private Sub set_pw(u_num As Integer)
' *** установка пароля ***
If Len(TextBox2.Text) = 0 Then
MsgBox "Введите Хэш"
Exit Sub
```

```

End If
hesh_x = TextBox2.Value
StrBuf = TextBox1.Text
sbL = Len(StrBuf)
If sbL < 3 Then
MsgBox "Плохой пароль!" & vbCrLf & "Повторите ввод"
Exit Sub
End If
pw(u_num, 0) = sbL
For i = 0 To sbL - 1
pw(u_num, i + 1) = Asc(Mid(StrBuf, i + 1, 1)) Mod
hesh_x
Next i
Call pw_show
End Sub
Private Sub check_pw(u_num As Integer, f_name As
String)
'*** проверка пароля ***
Dim u_name As String
Dim u_pw As String
Dim u_pw_hesh(20) As Integer
Dim StrBuf As String

Dim path_x As String
Dim hFileIn As Long
Dim buf_info() As String
Dim a As Integer, b As Integer
Dim str_log As String
path_x = ThisWorkbook.Path & "\"
u_name = ComboBox1.Value
str_log = u_name + " "
u_pw = TextBox1.Text
If Len(u_pw) <= 3 Then
'*** Введен неверный пароль ***
str_log = str_log & " Длина пароля <=3 " & Now()
try_pwd(try_pwd_ind) = str_log
try_pwd_ind = try_pwd_ind + 1
Call try_pwd_into_list
Exit Sub
End If
StrBuf = Dir(path_x + f_name)
If Len(StrBuf) = 0 Then
'*** Нет файла паролей ***
str_log = str_log & " Нет файла паролей " & Now()
try_pwd(try_pwd_ind) = str_log
try_pwd_ind = try_pwd_ind + 1

```



```

Call try_pwd_into_list
Exit Sub
End If
hFileIn = FreeFile
Open path_x + f_name For Input Access Read As #hFileIn
Do Until EOF(hFileIn)
Line Input #hFileIn, StrBuf
p = InStr(1, StrBuf, u_name)
If p > 0 Then
buf_info = Split(Trim(StrBuf), " ")
If Len(u_pw) <> UBound(buf_info) - 1 Then
'*** уточнить пароль ***
str_log = str_log & " Пароль не совпал по длине " &
Now( )
try_pwd(try_pwd_ind) = str_log
try_pwd_ind = try_pwd_ind + 1
Call try_pwd_into_list

Close hFileIn
Exit Sub
End If
For i = 2 To UBound(buf_info)
a = Asc(Mid(u_pw, i - 1, 1)) Mod hesh_x
b = buf_info(i)
If a <> b Then
'*** уточнить пароль ***
str_log = str_log & " Пароль не совпал по сим-волам " &
Now( )
try_pwd(try_pwd_ind) = str_log
try_pwd_ind = try_pwd_ind + 1
Call try_pwd_into_list
Close hFileIn
Exit Sub
End If
Next i
MsgBox "All OK"
'*** Успешный пароль ***
str_log = str_log & " Успешный пароль " & Now( )
try_pwd(try_pwd_ind) = str_log
try_pwd_ind = try_pwd_ind + 1
Call try_pwd_into_list
Exit Do
End If
Loop
Close hFileIn
End Sub

```

```

Private Sub try_pwd_into_list()
ListBox1.Clear
For i = 0 To try_pwd_ind
ListBox1.AddItem try_pwd(i)
Next
End Sub
Private Sub CommandButton1_Click()
'*** set password ***
Dim sbL As Integer
Dim StrBuf As String

Dim numP As Integer
numP = ComboBox1.ListIndex
Call set_pw(numP)
End Sub
Private Sub CommandButton2_Click()
'*** Check password ***
Call check_pw(ComboBox1.ListIndex, "pwd.txt")
End Sub
Private Sub CommandButton3_Click()
'*** exit ***
Unload UserForm2
End Sub
Private Sub CommandButton4_Click()
'*** PW into FILE ***
Call file_pw("pwd.txt")
End Sub
Private Sub UserForm_Initialize()
TextBox1.PasswordChar = "*"
hesh_x = 31
TextBox2.Value = hesh_x
Names(0) = "Иванов С.А."
Names(1) = "Петров В.Г."
Names(2) = "Сидоров С.С."
For i = 0 To 2
ComboBox1.AddItem Names(i)
Next
ComboBox1.ListIndex = 0
For i = 0 To 200
try_pwd(i) = ""
Next
try_pwd_ind = 0
End Sub
Sub pw_show()
Dim strbuf1 As String
Dim i As Integer

```

```

ListBox1.Clear

For i = 0 To 2
    strbuf1 = Names(i) + " "
    For j = 1 To pw(i, 0)
        strbuf1 = strbuf1 & pw(i, j) & " "
    Next j
    strbuf1 = strbuf1 & " *** " & pw_preobr(i)
    ListBox1.AddItem strbuf1
Next i
End Sub

Function pw_preobr(ind As Integer) As String
    Dim sbuf As String
    For i = 1 To pw(ind, 0)
        sbuf = sbuf & Chr(pw(ind, i))
    Next
    pw_preobr = sbuf
End Function

Function is_hesh() As Boolean
    For i = 0 To 2
        If pw(i, 0) = 0 Then
            is_hesh = False
            Exit Function
        End If
    Next i
    is_hesh = True
End Function

Private Sub file_pw(fname As String)
    Dim hFile As Long
    Dim path_x As String
    Dim StrBuf As String
    path_x = ThisWorkbook.Path + "\"
    If Not is_hesh Then
        MsgBox "Все пароли должны быть введены"
    End If
    Exit Sub
End Sub

hFile = FreeFile
Open path_x + fname For Output Access Write As #hFile
For i = 0 To 2

    StrBuf = Names(i) & " "
    For j = 1 To pw(i, 0)
        StrBuf = StrBuf & pw(i, j) & " "
    Next j
    Print #hFile, Trim(StrBuf)
Next i

```

```
Close #hFile  
MsgBox "Пароли сохранены успешно"  
End Sub
```