

Számítástechnika

Szerveetlen és Analitikai Kémia Tanszék

Tanszékvezető: *Dr. Nyulászi László*

Tárgy előadója: *Dr. Simon András (Ch ép. fszt. 33., 35.)*

email: andras.simon@mail.bme.hu, sim1971@gmail.com

tel: 4632293, 4633411

Tárgy rendszergazda: *Rippel Endre (Ch C2)*

<http://www.kit.bme.hu>

1

A Számítástechnika c. tárgy beosztása 2018/2019/I. tanévben

Hét	Héttartam	Ködt előadás	Hétfő gyakorlat	Ködt gyakorlat	Szerda gyakorlat	Csütörtök gyakorlat	Szombat gyakorlat	Előadás: Minden tanlóre kedd, 14-15	Gyakorlatok oktatói	Helyettes oktatói
		CH-C14	CH gépt.	CH gépt.	CH gépt.	CH gépt.	CH gépt.			
1	szept. 03-07.	Bücs mind	Bücs mind	Bücs 3	Bücs 1	Bücs 4,5	Bücs 2			
2	szept. 10-14.	előadás-1	AB	AB	AB	AB	AB	1: Kedd, 17-19: Jägeróki Gyula		Simon András
3	szept. 17-21.	előadás-2	AB	AB	AB	AB	AB	2: Hétfő, 14-16: Rippel Endre		Szűcs Róza
4	szept. 24-28.	előadás-3	AB	AB	AB	AB	AB	3: Hétfő, 16-18: Szűcs Róza		Rippel Endre
5	okt. 01-05.	előadás-4	AB	AB	AB	AB	AB	4: Szerda, 10-12: Rippel Endre+Kovács Adá		Gömbs Péter
6	okt. 08-12.	előadás-5	AB	AB	AB	AB	AB	5: Kedd, 15-17: Simon András		Gömbs Péter
7	okt. 15-19.	konzultáció	AB	AB	AB	AB	AB	Biológus		
8	okt. 22-26.	konzultáció	AB	AB	AB	AB	AB	1: Kedd, 08-10: Gömbös Péter		Rippel Endre
9	okt. 29. nov. 02.	órápótlás	AB	AB	AB	AB	AB	2: Csütörtök, 14-16: Jägeróki Gyula		Rippel Endre
10	nov. 05-09.	ZH-1	AB	AB	AB	AB	AB	3: Hétfő, 10-12: Rippel Endre		Szűcs Róza
11	nov. 12-16.	konzultáció	AB	AB	AB	AB	AB	4: Szerda, 12-14: Simon András		Rippel Endre
12	nov. 19-23.	konzultáció	AB	AB	AB	AB	AB	5: Szerda, 14-16: Szűcs Róza		Simon András
13	nov. 26-30.	konzultáció	AB-pótlás	AB-pótlás	AB	AB	AB			
14	dec. 03-07.	ZH-2	AB-pótlás	AB-pótlás	AB	AB	AB	Orvostudomány		
15	dec. 10-14.	pótlás	pótlás	pótlás	AB-pótlás	AB-pótlás	AB-pótlás	08-11:9: előzetes (1)		
16	dec. 17.	pótlás	pótlás	pótlás	AB-pótlás	AB-pótlás	AB-pótlás	12:0-16:9: előzetes (2)		
								17:0-21:9: közpess (3)		
								22:0-26:9: j6 (4)		
								27:0-33:6: j6 (5)		

- Lehet, hogy 2 ember lesz egy gép előtt.
- Feladatok teremtőpontokat (előadás) nem adjuk le, konzultációs alkalmakra, vagy pótlásokra szolgálnak.
- Előzetes megbeszélésük valamelyik este 19:00 (7 óra) szerdai a gépteremben. Ha kevesen jelennek meg, szóbeli elővona lesz.
- A hallgatóknak előre jelentkezni kell majd (a gyakorlatvezetői mailben kérem sokszor továbbítani).
- Az órák saját belátásuk szerint adhatnak az órákon pluszpontokat (max. 3). Pluszpont járhat, ha akkor is bejön a pótlásra dolgozni a hallgató, ha nem kellene, mert megvan az előtti katalógusozás.
- A hallgatók három labor mulasztással pótlás nélkül.
1. sz. október 30., 2. sz. december 4., mindkét sz CH C14 és helyszíni pótlás: december 11. 14:00 helyszíni, pótlás (90 perc, 30 pont) december 17. 10:00 helyszíni.
- Az elő- és utóévek sz-uk külön-külön legalább 5 ponttal kell lenni az előzetesnél jobb osztályzat megszerzéséhez.
- A pótlás automatikusan felülírja a rosszabbik sz eredményét, akkor is, ha kevesebb pontot ért el a hallgató, vagy esetleg még is bakt.

1. Gyakorlatokon való részvétel

Csak három labor mulasztható. Kivétel, aki ZH-eredmények miatt ismételt, nekik a részvétel csak ajánlott. Félév végén egy labor pótlása lehetséges.

2. ZH-eredmény

Az elérhető 15-ből legalább 5 pont, illetve összesen legalább 12 pont megszerzése. Utolsó héten pót-ZH pótlható, eredménye a gyengébb ZH eredményét felülírja.

2

Ajánlott irodalom:

Kovalcsik Géza: Az Excel programozása, ComputerBooks, Budapest, 2008, ISBN 963-618-332-5

Balogh Gábor: Visual Basic és Excel programozás, ComputerBooks, Budapest, 2002, ISBN 963-618-229-9

Kuzmina Jekatyerina, Dr. Tamás Péter, Tóth Bertalan: Programozunk Visual Basic rendszerben, ComputerBooks, Budapest, 2006, ISBN 963-618-308-2

Billo E. Joseph: Excel® for Chemists: A Comprehensive Guide, Wiley-VCH, New York – Chicester – Weinheim – Brisbane – Singapore – Toronto, 2001, ISBN 0-471-39462-9 (Paperback), 0-471-22058-2 (Electronic)

Billo E. Joseph: Excel for scientists and engineers, John Wiley & Sons, Inc., Hoboken, New Jersey, ISBN: 978-0-471-38734-3

3

Mivel foglalkozunk a laborokon?

Excel alapok, cellahivatkozás, munkalapfüggvény, diagramvarázsló, trendvonal, Solver

VBA alapok, típus, értékadás, kiíratás cellába, beolvasás billentyűről, If - Then , Select Case, If - Then - Else Go To, Do - Loop, programírás blokkdiagramról

For - Next (Sorozat elemeinek összegzése); Function

Tömbök, beolvasás fájlból, adatok file-ba írás (Skalársorozat; Vektor hossza; két vektor szöge; max, min elem)

Szubrutin írás, makró rögzítése

Word, Chemscketch

4

A gyakorlat célja: adatkezelésben illetve programozásban alapismeretek és elemi jártasság megszerzése, későbbi félévek során a laborokban a jegyzőkönyvek elkészítéséhez való készségek megszerzése

Operációs rendszerek:

Windows
Mac OS
Linux
Unix
VMS

Szoftverek:

MS Office
(Open Office) Word
 Excel
 Power Point

Rasmol
Chemsketch
Chemdraw
Origin
Isis Draw
Chemwindow
Hyperchem

Programnyelvek:

Visual Basic (VBA)
Fortran
Turbo Pascal
C
C++

Programozás

- utasítások pontosan, sorrendben végrehajtandó sorozata
- pontosan olyan, mintha egy könyvet olvasnánk, amiben le van írva, hogy sorjában mit kell tennünk éppen most, beleértve, hogy hova lapozzunk a következő teendő elolvasásához
- célja hosszabb számítások elvégzése ill. kiírítása
- akkor érdemes programot írni, ha sok számolni való van illetve ha ugyanazt az (esetleg rövid és egyszerű) számítást akarjuk sokszor elvégeztetni – (pl. sokrészecske probléma)
- a programozás filozófiáját érdemes megtanulni
- egy új programnyelvet mindig sokkal könnyebb megtanulni, ha az ember ismer már egy másikat
- programnyelv hatékony tanulása – gyakorlattal !!!

A programozás elemei

- változók, tömbök (típus, deklaráció)
- értékadás, műveletek (változó típustól függ)
- beolvasás, kiírás, elmentés
- ciklus utasítás
 - egyszerű ciklus
 - feltételes ciklus
- feltételes utasítás
- függvény hívása
 - külső függvény
 - belső függvény
- eljárás (szubrutin) hívása

```

Rem (Declarations)
' Fordítási opciók beállítása
Option Explicit
Option Base 1
Option Compare Text
Option Private Module

' Modul szintű deklarációk
Dim, Private, Public, Const, Type

Rem Eljárás
Sub Eljárás(paraméterek)
' Lokális deklarációk
' Végrehajtható utasítások
End Sub

Rem Függvény(paraméterek)
Function Függvény(paraméterek) As Függvénytípus
' Lokális deklarációk
' Végrehajtható utasítások
Függvény = Függvényérték ' a függvényérték beállítása
End Function

2.3. ábra
A Visual Basic programmodulok általános felépítése

Rem modul szintű deklarációk:
Option XXXX ' fordítási beállítások
[Public | Private] Const ' konstans-deklarációk
[Public | Private] Type ' felhasználói típus
Public | Private | Dim ' változó-deklarációk

[Public|Private|Static] Sub eljárásnév(paraméterlista)
Const ' lokális konstans-deklarációk
Static | Dim ' lokális változó-deklarációk
[az eljárás törzseinek utasításai]
[Exit Sub]
Const ' lokális konstans-deklarációk
Static | Dim ' lokális változó-deklarációk
[az eljárás törzseinek utasításai]
End Sub

[Public|Private|Static]Function évnév(paraméterlista) As típus
Const ' lokális konstans-deklarációk
Static | Dim ' lokális változó-deklarációk
[az eljárás törzseinek utasításai]
Évnév = kifejezés ' a függvény értéke
[Exit Function]
Const ' lokális konstans-deklarációk
Static | Dim ' lokális változó-deklarációk
Évnév = kifejezés ' a függvény értéke
End Function

```

Egy „programocska”

A blokkdiagram:

A VBA program:

```

Sub kobok()
Cells(1, 1) = "x"
Cells(1, 2) = "x köbe"
x = 2
ujra:
Cells(x, 1) = x
Cells(x, 2) = x ^ 3
x = x + 1
If x <= 8 Then GoTo ujra
End Sub

```

	A	B
1	x	x köbe
2	2	8
3	3	27
4	4	64
5	5	125
6	6	216
7	7	343
8	8	512
9		

Amit a program kiír

9

Egy „programocska”

VBA program felépítése:

```

Sub <neve> ( )
<deklarációk>
<utasítások>
End Sub

```

A deklaráció a változók típusát és dimenzióját adja meg

egyes deklarációk elválasztása: vesszővel vagy új utasításként

utasítások elválasztása: kettősponttal vagy új sorba írva

Sub kobok()

```

Cells(1, 1) = "x"
Cells(1, 2) = "x köbe"
x = 2
ujra:
Cells(x, 1) = x
Cells(x, 2) = x ^ 3
x = x + 1
If x <= 8 Then GoTo ujra
End Sub

```

	A	B
1	x	x köbe
2	2	8
3	3	27
4	4	64
5	5	125
6	6	216
7	7	343
8	8	512
9		

Amit a program kiír

10

Program végrehajtása

```

Sub kobok()
Cells(1, 1) = "x"
Cells(1, 2) = "x köbe"
x = 2
ujra:
Cells(x, 1) = x
Cells(x, 2) = x ^ 3
x = x + 1
If x <= 8 Then GoTo ujra
End Sub

```

	A	B
1	x	x köbe
2	2	8
3	3	27
4	4	64
5	5	125
6	6	216
7	7	343
8	8	512
9		

11

Változók

- A program az adatokat **változókban** tárolja
- pl. ha $a=2$, akkor $x=a^3-15 = -7$
- vagy, pl. ha $b=8$, akkor $y=3*b/2 = 12$
- a számítógép az a , b , vagy bármely más **változót** mindig az **értékével** helyettesíti
- akkor is, ha **nem** adtunk neki értéket, csak **akkor nem tudjuk, milyen értékkel...**

12

Tömbök

- A tömb **összetartozó változók együttese** (pl. vektor koordinátái, mátrix elemei...)
- ugyanaz a változónév jelöli a tömb egyes elemeit, a megkülönböztetést a tömb indexe(i) jelenti

Tömb deklarálása: **Dim utasítás**

Dim változónév(max_index1[, max_index2...])

Dim Nevek(30) as string * 20
 Dim Matrix(4,3) as single
 Dim Reszlet%(3 To 5)
 Dim s\$(100), y(100) As Boolean

} Option Base 1

17

Tömbök

Dim Matrix() as single
 Dim intSzam1%, intSzam2 as integer

...

intSzam1 = 5

intSzam2 = 3

ReDim Matrix(intSzam1,intSzam2) ← Dinamikus tömb

...

intSzam1 = 4

intSzam2 = 4

ReDim Matrix(intSzam1,intSzam2)

vagy

ReDim Preserve Matrix(5,intSzam2)

} Nem ugyanaz
 A hatás!

18

Felhasználói adattípus (Rekordtípusú változó)

Type Dolgozok

Veznev as string * 30

Kernev as string * 20

SzulDate as Date

AnyjaNeve as string * 40

BelepDate as Date

Fizetes as single

End Type

...

Dim Alkalm(100) as Dolgozok

...

Alkalm(1).Veznev = "Kovacs"

Alkalm(1).Belep = #1/1/2003#

VAGY:

Dim Adatok1(100,3) as string

Dim Adatok2(100,2) as date

Dim Adatok3(100) as single

19

<i>Date érték</i>	<i>Dátum</i>	<i>Idő</i>
0	1899.12.30.	0:00:00
1.5	1899.12.31.	12:00:00
-18917.25	1848.03.15.	6:00:00
29212.75	1979.12.23.	18:00:00

Objektumok elnevezése

Dim objA as object

Set objA = Workbooks("Munkafuzet1").Worksheets(1).Cells(1,1)

Dim rngA as range

Set rngA = Workbooks(1).Worksheets(1).Range("A1 ")

objA = "Alma"

vagy

rngA = "Alma"

20

Értékkadás

Az = utasítás segítségével ("legyen egyenlő")
 változó = érték

n = 15
 s = "Ez a szöveg"
 a = b/2
 x = x+1
 s = False
 z = True
 q = x <= 4

Az = nem pontosan azt jelenti, amit matematikában megszoktunk, a két oldal most nem felcserélhető!

x = 15	x = 15
y = 3	y = 3
x = y	y = x

most x és y értéke is 3 !
 most x és y értéke is 15 !

21

Műveletek

- aritmetikai műveletek: + - * / \ ^
 Integer
 Long
 Single
 Double

- logikai műveletek: And Or Not
 Boolean

- string műveletek: +
 String

aritmetikai műveletek

- x = 1+2
- x = 18-2*y
- x = (18-2)*y
- x = y/24
- x = x+y
- x = y^3

logikai műveletek

- x = y Or z
- x = y And z
- x = Not y
- x = Not y And z
- x = Not (y And z)

string műveletekha
 s="osztogat",
 akkor s="f"+ s után
 s értéke „fosztogat”

22

Művelet	Operátor	Példa	Eredmény
HATVÁNYOZÁS	^	3^2	9
SZORZÁS	*	2*4	8
OSZTÁS	/	7/3	2,3333
Egészosztás	\	7\3	2
MARADÉKOSZTÁS	Mod	7 Mod 3	1
ÖSSZEADÁS	+	7+4	11
KIVONÁS	-	7-4	3

Operátor	Példa	How olvassuk a kifejezést?	Eredmény
=	7 = 3	Igaz az állítás, hogy 7 egyenlő 3-al?	False (hamis)
>	7 > 3	Igaz az állítás, hogy 7 nagyobb mint 3?	True (igaz)
<	7 < 3	Igaz, hogy 7 kisebb mint 3?	False (hamis)
>=	7 >= 3	Igaz, hogy 7 nagyobb vagy egyenlő mint 3?	True (igaz)
<=	7 <= 3	Igaz, hogy 7 kisebb vagy egyenlő mint 3?	False (hamis)
<>	7 <> 3	Igaz, hogy 7 nem egyenlő 3-al?	True (igaz)

23

függvényhívás	leírás	az argumentum	a függvényérték típusa
abs(x)	x	egész, valós	egész, valós
atn(x)	arc tg x	egész, valós	Double
cos(x)	cos x	egész, valós	Double
exp(x)	e ^x	egész, valós	Double
fix(x)	[x] egészrész	egész, valós	egész, valós
int(x)	egész ≤ x	egész, valós	egész, valós
log(x)	ln x	egész, valós	Double

függvényhívás	leírás	az argumentum	a függvényérték típusa
rnd()	{0, 1} intervallumba eső véletlen szám (indítást: randomize())		Single
round(x[,n])	n tizedes jegyre kerekítő függvény	egész, valós	egész, valós
sgn(x)	előjelfüggvény	egész, valós	egész, valós
sin(x)	sin x	egész, valós	Double
sqr(x)	√x	egész, valós	Double
tan(x)	tg x	egész, valós	Double
π meghatározása	4*atn(1)		
e meghatározása	exp(1)		

2.10. ábra
 Matematikai függvények

Aritmetikai műveletek	Összehasonlító műveletek	Logikai műveletek
hatványozás (^)	egyenlő (=)	tagadás (Not)
előjelváltás (-)	nem egyenlő (<>)	és (And)
szorzás, osztás (*, /)	kisebb (<)	vagy (Or)
egész osztás (\)	nagyobb (>)	kizáró vagy (Xor)
osztási maradék (mod)	kisebb vagy egyenlő (<=)	ekvivalencia (Eqv)
összeadás kivonás (+, -)	nagyobb vagy egyenlő (>=)	implikáció (Imp)
sztringek összefűzése (&)	hasonló (like), megegyezik (is)	

2.9. ábra
A Visual Basic operátorok precedenciája

Karakter	Mit helyettesít?
*	Egy vagy több karaktert helyettesít.
?	Egyetlen szöveg vagy szám karaktert helyettesít.
[...]	Egy karakter helyre helyettesíti a zárójelek közé írt betűket, vagy számokat, vagy a kötőjelekkel írt karakterek tartományát.
[!...]	Egy karakter helyre kihagyja a zárójelek közé írt betűket, vagy számokat, vagy a kötőjelekkel írt karakterek tartományát.
#	Egy pozícióra bármilyen szám helyettesítésére szolgál.

25

Sub Osszehasonlitas	Eredmény
<code>Debug.Print "Motor" Like "M"</code>	True
<code>Debug.Print "Motor" Like "?o"</code>	True
<code>Debug.Print "3-as" Like "#-as"</code>	True
<code>Debug.Print "u-as" Like "#-as"</code>	False
<code>Debug.Print "3-as" Like "?-as"</code>	True
<code>Debug.Print "u-as" Like "?-as"</code>	True
<code>Debug.Print "Jani" Like "Jan[ió]"</code>	True
<code>Debug.Print "Jani" Like "[A-M]"</code>	True
<code>Debug.Print "Jani" Like "[!A-M]"</code>	False

End Sub

Option Compare Text

26

Operátor	Mikor True (igaz) az eredmény?
And (és)	Az eredmény akkor és csak akkor True, ha egyszerre egy időben az összes vizsgált érték True. Vagyis, ha mind bolX, mind bolY tartalma True.
Or (vagy)	A Vagy logikai művelet minden esetben True (igaz) értéket ad vissza, ha bármelyik vizsgált érték True (igaz) értéket tartalmaz. Máshogy fogalmazva vagy ennek vagy annak, de legalább egyiknek igaznak kell lennie.
Xor (kizáró vagy)	Ha az Xor műveletet végzünk, akkor annak az eredménye akkor ad True értéket eredményül, ha a vizsgált változók nem azonos értékeket tárolnak. Vagyis, ha az összes vizsgált érték True vagy mind False, akkor az eredmény False, egyébként True.
Eov (ekvivalencia)	Az Xor művelet párja az Eqv. Ez a művelet az egyezőséget vizsgálja. Ha az összes változó vagy True vagy False érték, az eredmény True, egyébként False.
Not (negálás)	Ez egy logikai változó tartalmát váltja át ellenkező értékűre. Ha egy logikai érték elé írjuk, akkor ha az True volt, False eredményt ad, ha False volt az eredmény True lesz.

27

Beolvasás, kiírás

- Beolvasás Excel cellából / kiírás Excel cellába
- Beolvasás fájlból / kiírás fájlba
- Beolvasás InputBox segítségével

1. Cellaműveletek:

az Excel cellákra
mint változókra
hivatkozhatunk

	A	B	C
1			
2			
3			

Cells(1,1) - A1
Cells(1,2) - B1
Cells(2,1) - A2

Fordított a sorrend
mint az Excel-ben !

cellahivatkozás:

Cells(sor száma, oszlop száma)

x = Cells(1,3)

y = 5 + 3*Cells(2,5)

Cells(3,2) = x + 3*y

Cells(1,1) = Not b

Cells(2,4) = 3*Cells(1,4)

2. Olvasás fájlból / írás fájlba

Open fájlnev For megnyitás típusa As #I/O csatorna száma

Input #I/O csatorna száma beolvasandó változók
Write # I/O csatorna száma kiírandó változók

Close # I/O csatorna száma

fájl neve ""-ben
 vagy szöveg változó

Input
Output
Append

egész szám vagy változó
 ez különbözteti meg a fájlokat
 ha több is nyitva van egyszerre

Open "vektorok.dat" For Input As #1
Open "osszegvektor.dat" For Output As #2
Input #1, v1(1), v1(2), v1(3)
Input #1, v2(1), v2(2), v2(3)
Write #2, v1(1)+v2(1), v1(2)+v2(2), v1(3)+v2(3)
Close #1
Close #2

vektorok.dat:
 2.1 5.3 1.5
 1.0 2.2 1.5
összegvektor.dat:
 3.1 7.5 3.0
 29

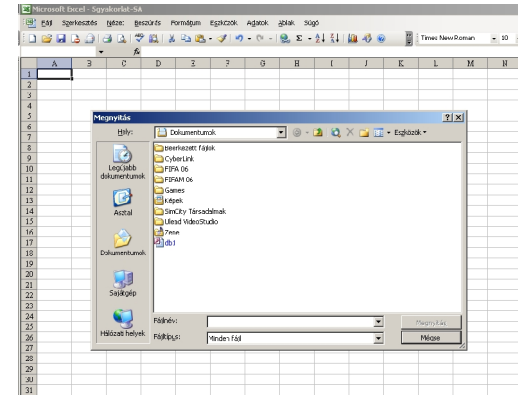
Dim fnev As String

fnev = Application.GetOpenFilename

Open fnev For Input As #1

Input #1, intN

Close #1



3. Beolvasás InputBox-szal

változónak adhatunk értéket az **InputBox** utasítással is, a kívánt értéket ekkor a billentyűzetről olvassuk be

InputBox (Prompt [,Cím, Default, Xpos, Ypos])

Szövegek vagy szöveg változók
 az InputBox "magyarázatait"

Az InputBox helye a képernyőn

A beolvasandó változó
 alapértelmezett értéke



x = InputBox("Adja meg x értékét", "Beolvasás", "Ide kell írni", 5000, 5000)

Visual Basic for Excel – elméleti összefoglaló

Változók típusai, kifejezések, értékadás, feltételes utasítás

Néhány alaptípus:

Integer(%) Double(#) Single(!) String(\$) Boolean

Típus deklarációja nem kötelező DE erősen ajánlott!

Alapműveletek: + - * / \ ^ (aritmetikai)

And Or Not (logikai)

Reláció jelek: = < > <= >= <>

Aritmetikai kifejezés: a*a*a - 81

Logikai kifejezés: fa*fm<0

Értékadás: változó = érték (kifejezés)

Pl: fa = a^4 - 81

VBA program felépítése:

Sub <neve> ()
 <esetleg deklarációk>
 <utasítások>
End Sub

Ciklus utasítás

Értelme – ha ugyanazt a műveletsort sokszor, más-más értékekkel is el kell végezni

- For...Next
- Do While...Loop
- Do Until...Loop
- Do ... Loop While
- Do...Loop Until
- If ... Goto

1. For...Next ciklus

For ciklusváltozó = induló érték To végérték [Step lépésköz]

utasítások

Next ciklusváltozó

For i=1 To 5

Cells(1,i) = i

Next i

For i = 1 To 5 Step 2

Cells(3,i) = i

Next i

For i = 5 To 1 Step -1

Cells(2,i) = -i

Next i

	A	B	C	D	E
1	1	2	3	4	5
2	-1	-2	-3	-4	-5
3	1		3		5
4					

33

2. Do...Loop ciklusok

Do While logikai kifejezés

utasítások

Loop

Do Until logikai kifejezés

utasítások

Loop

Ciklus elején értékelődik ki

Akkor hajtódik végre a ciklus, ha **igaz**

Akkor hajtódik végre a ciklus, ha **hamis**

Ciklus végén értékelődik ki

Do

utasítások

Loop While logikai kifejezés

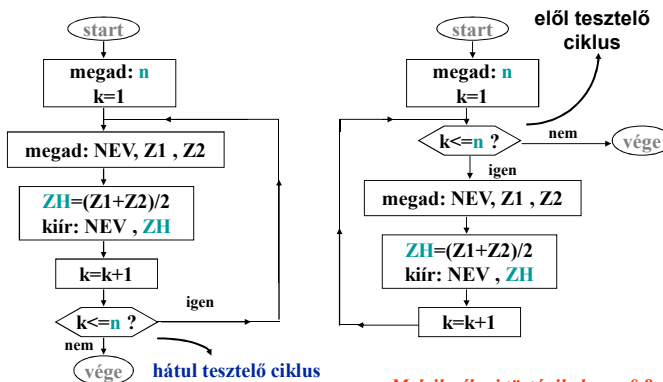
Do

utasítások

Loop Until logikai kifejezés

Ciklusok

Két ZH átlagának kiszámítása n hallgató esetén.



Melyiknél mi történik, ha n=0 ?

35

Ciklusok

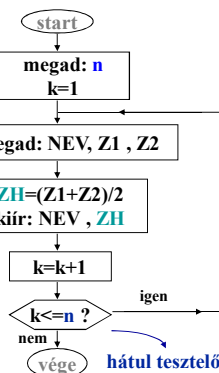
Két ZH átlagának kiszámítása n hallgató esetén.

Do - Loop While ciklus

Visual Basic program részlet

```

n=InputBox("n=?"): k=1
Do
  NEV=InputBox("NEV=?")
  Z1=InputBox("Z1=?")
  Z2=InputBox("Z2=?")
  ZH=(Z1+Z2)/2 : Cells(k,1)=NEV
  Cells(k,2)=ZH : k=k+1
Loop While k<=n
    
```



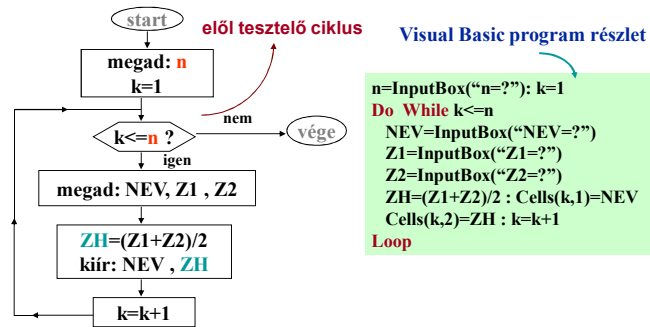
hátsó tesztelő ciklus

36

Ciklusok

Két ZH átlagának kiszámítása n hallgató esetén.

Do While - Loop ciklus



37

Ciklusok

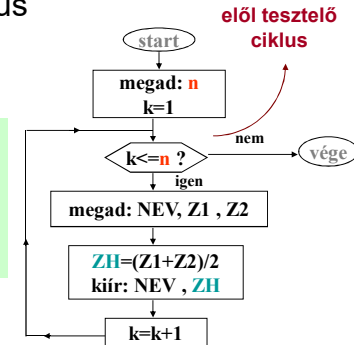
Két ZH átlagának kiszámítása n hallgató esetén.

For To - Next ciklus

Visual Basic program részlet

```

n=InputBox("n=?")
FOR k=1 TO n
  NEV=InputBox("NEV=?")
  Z1=InputBox("Z1=?")
  Z2=InputBox("Z2=?"): ZH=(Z1+Z2)/2
  Cells(k,1)=NEV : Cells(k,2)=ZH
NEXT k
  
```



38

For Each - Next ciklus

```

Dim rngCell As Range
Dim rngTart As Range
Set rngTart = Range("A1:F21")
  
```

```

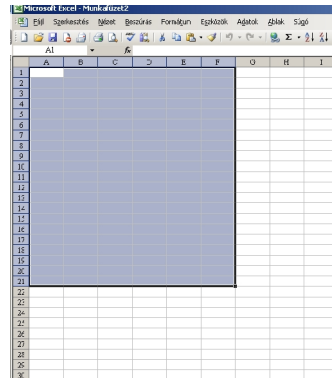
For Each rngCell In rngTart.Cells
...
Next rngCell
  
```

```

Range("A1").Select
  
```

```

...
For Each rngCell In Selection.CurrentRegion
  
```



39

Feltételes utasítás

Különböző esetekben más-más utasítás hajtandó végre

```

If logikai kifejezés Then
  utasítás1
Else
  utasítás2
End If
  
```

Akkor hajtódik végre ha a logikai kifejezés igaz

Akkor hajtódik végre ha a logikai kifejezés hamis

utasítás1 ill. utasítás2 lehet összetett utasítás (több utasítás együttese) is

Feltételesen végrehajtandó utasítás:

```

If logikai kifejezés Then egyszerű utasítás1
  
```

```

If logikai kifejezés Then
  összetett utasítás1
End If
  
```

(ha nincs Else ág)

40

Többfelé ágazó feltételes utasítás:

```

If logikai kifejezés 1 Then
    utasítás1
Elseif logikai kifejezés 2 Then
    utasítás2
Else
    utasítás3
End If
    
```

Akkor hajtódik végre, ha

- logikai kifejezés 1 igaz
- logikai kifejezés 1 hamis, DE logikai kifejezés 2 igaz
- mindkét logikai kifejezés hamis

utasítás3 helyén természetesen újabb feltételes utasítás is állhat !

41

```

If logikai kifejezés 1 Then
    utasítás1
Else
If logikai kifejezés 2 Then
    utasítás2
Else
If logikai kifejezés 3 Then
    utasítás3
Else
If logikai kifejezés 4 Then
    utasítás4
End If
End If
End If
End If
    
```

```

Select Case változó
    Case érték1
        utasítás1
    Case érték2
        utasítás2
    Case érték3
        utasítás3
End Select

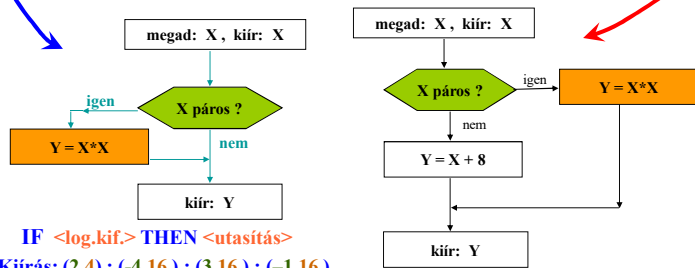
Case 1, 2, 3, 9
Case 4 To 8, Is < 1
Case Is > 10
    
```

42

Visual Basic for Excel – elméleti összefoglaló

feltélesen végrehajtandó utasítás és feltételes utasítás

Az X értékek (mindkét esetben): 2, -4, 3, -1



IF <log.kif.> **THEN** <utasítás>
 Kiírás: (2,4) ; (-4,16) ; (3,16) ; (-1,16)

IF <log.kif.> **THEN** <ut.1> **ELSE** <ut.2>
 Kiírás: (2,4) ; (-4,16) ; (3,11) ; (-1,7)

43

Függvények

Külső függvények:

függvény hívása a főprogramban

kifejezés = függvénynév (arg1,...arg_i)

Function függvénynév (f_arg1,...f_arg_i) **As típus**

.

függvénynév = kifejezés

End Function

függvény

Belső függvények:

olyan függvények amit a VBA "ismer", nem kell megírunk, lehet rögtön hívni

Abs(arg1) **Sin**(arg1) **Cos**(arg1) **Sqr**(arg1)

44

függvényhívás	leírás	az argumentum	a függvényérték típusa
<i>abs(x)</i>	x	egész, valós	egész, valós
<i>atn(x)</i>	arc tg x	egész, valós	Double
<i>cos(x)</i>	cos x	egész, valós	Double
<i>exp(x)</i>	e ^x	egész, valós	Double
<i>fix(x)</i>	[x] egészrész	egész, valós	egész, valós
<i>int(x)</i>	egész ≤ x	egész, valós	egész, valós
<i>log(x)</i>	ln x	egész, valós	Double

függvényhívás	leírás	az argumentum	a függvényérték típusa
<i>rnd()</i>	[0, 1]intervallumba eső véletlen szám (indítást: <i>randomize()</i>)		Single
<i>round(x[,n])</i>	n tizedes jegyre kerekítő függvény	egész, valós	egész, valós
<i>sgn(x)</i>	előjelfüggvény	egész, valós	egész, valós
<i>sin(x)</i>	sin x	egész, valós	Double
<i>sqr(x)</i>	√x	egész, valós	Double
<i>tan(x)</i>	tg x	egész, valós	Double
<i>π meghatározása</i>	4*atn(1)		
<i>e meghatározása</i>	exp(1)		

2.10. ábra
Matematikai függvények

Eljárások

eljárás hívása a főprogramban

Call eljárásnév (arg1,...arg_i)

Sub eljárásnév (s_arg1,...s_arg_i)

·

·

End Sub

eljárás

Eljárásnak kimenő és bemenő változói is lehetnek
 Függvénynek csak bemenő változói – az egyetlen kimenő érték most a függvény maga

46

Függvények használata

Példa: A [0;50] intervallumot *n=100* részre osztva *h=0,5* lépéssel minden *x* (belső) osztópontozhoz kiszámítandó *f(x)* és $\frac{f(x+h) - f(x-h)}{2h}$

A feladatot az alábbi függvényre kell elvégezni:

$$f(x) = x^2 + 2,3x + 8 \cos(x + 1) - \frac{x}{x + 2}$$

Ha nem használunk *Function*-t:
 a formulát **háromszor** be kell írni, rendre az *x*, *x+h*, *x-h* argumentumokkal

47

Függvények használata

Function f(x#) As Double
 $f = x^2 + 2.3 * x + 8 * \cos(x + 1) - x / (x + 2)$
End Function

A függvénynek típusa van: "As Double"

Azonosítójának értéket kell adni: "f = ..."

```

Sub szelo()
Dim n%, h#, x#, i%
n = 100: h = 0.5
x = 0
For i = 1 To 100
x = x + h
Cells(i, 1) = x
Cells(i, 2) = f(x)
Cells(i, 3) = (f(x + h) - f(x - h)) / (2 * h)
Next i
End Sub

```

Függvényhívást kifejezés helyére lehet írni

48

Egy Function-nak vagy Sub-nak lokális változói is lehetnek

Öt vektor mindegyikét szeretnénk megszorozni a hatodik vektorral...

```
Function skal(sor%, x#(), y#(), n%) As Double
Dim sum#, i%
sum = 0
For i = 1 To n
    sum = sum + x(sor, i) * y(i)
Next i
skal = sum
End Function
```

```
Sub sokvektor()
Dim a#(5, 3), b#(3), c#(5), k%, j%
Open "adat.txt" For Input As #1
' Itt beolvassuk a és b elemeit '
Close #1
For j = 1 To 5
    Cells(j, 7) = skal(j, a, b, 3)
Next j
End Sub
```

Feladat megoldása
függvényvel

1	3	2				3
-2	0	6		1		4
1	1	1	•	0	=	2
0	0	5		1		5
-3	1	4				1

49

Egy Function-nak vagy Sub-nak lokális változói is lehetnek

Öt vektor mindegyikét szeretnénk megszorozni a hatodik vektorral...

```
Sub skal (sor%, x#(), y#(), n%, prod#)
Dim sum#, i%
sum = 0
For i = 1 To n
    sum = sum + x(sor, i) * y(i)
Next i
prod = sum
End Sub
```

```
Sub sokvektor()
Dim a#(5, 3), b#(3), c#(5), k%, j%, szorz#
Open "adat.txt" For Input As #1
' Itt beolvassuk a és b elemeit '
Close #1
For j = 1 To 5
    Call skal(j,a,b,3,szorj)
Cells(j, 7) = szorz
Next j
End Sub
```

Feladat megoldása
eljárással

1	3	2				3
-2	0	6		1		4
1	1	1	•	0	=	2
0	0	5		1		5
-3	1	4				1

50

De ha a vektorok abszolút értéke is kell akkor csak eljárással oldható meg!!

```
Sub skal (sor%, x#(), y#(), n%, prod#, xa#, ya#)
Dim sum#, sumx#, sumy#, i%
sum = 0: sumx = 0: sumy = 0
For i = 1 To n
    sum = sum + x(sor, i) * y(i)
    sumx = sumx + x(sor,i) * x(sor,i)
    sumy = sumy + y(i) * y(i)
Next i
xa = sqr(sumx): ya = sqr(sumy)
prod = sum
End Sub
```

```
Sub sokvektor()
Dim a#(5, 3), b#(3), c#(5), k%, j%, szorz#, aa#, bb#
Open "adat.txt" For Input As #1
' Itt beolvassuk a és b elemeit '
Close #1
For j = 1 To 5
    Call skal(j,a,b,3,szorj,aa,bb)
Cells(j, 7) = szorz
Cells(j, 8) = aa
Next j
Cells(5,9) = bb
End Sub
```

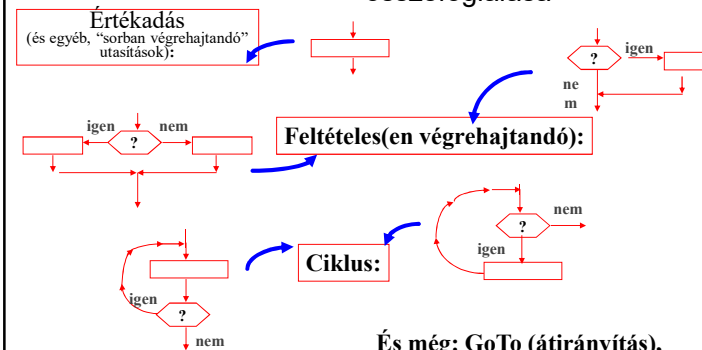
Feladat megoldása
eljárással

1	3	2				3
-2	0	6		1		4
1	1	1	•	0	=	2
0	0	5		1		5
-3	1	4				1

51

Visual Basic for Excel – elméleti összefoglaló

Az eddig tanult **UTASÍTÁSOK**
összefoglalása



És még: GoTo (átírányítás),
ADAT beolvasása és kiírása !!

52