




Törley Gábor (pezsgo@elte.hu)

# PROGRAMOZÁSI NYELVEK, KÖRNYEZETEK KIFEJEZŐEREJÉNEK VIZSGÁLATA



# Bevezetés

- Egy nyelv annál kifejezőbb, minél kevesebb szellemi befektetést, időt kíván a programozótól a kódolás során
  - Nyelv és környezet: komplex egység
  - Módszeres programozás lépései: Kódolás, tesztelés, hibakeresés és javítás
- 

# Bevezetés

- Nyelvi szempontok
  - Értelmes alapszavak
  - Egyszerű és következetes programszerkezet
  - Egyszerű kódolás, könnyen tanulhatóság
- Fejlesztői környezet szempontjai
  - Kulcsszavak kiemelése
  - Kódolást segítő szolgáltatások (pl. intelisence)
  - Fordítás, hibaüzenetek
  - Nyomkövető rendszer

# Bevezetés

- Előzetes tudás
  - Algoritmizálás alapjai
  - Imagine, Logo ismeret
- Imperatív nyelvek osztálya
- Gondolatkísérlet
  - Adott egy egészekből álló számsorozat, döntsük el, van-e köztük páros szám!
- Adott a specifikáció és az algoritmus

## Specifikáció:

Bemenet:  $N \in \mathbf{N}, \text{Tömb} \in \mathbf{Z}^*, \text{Páros}: \mathbf{Z} \rightarrow \mathbf{L}$  [ $\mathbf{L} = \{\text{igaz}, \text{hamis}\}$  – Logikai értékek halmaza]

Kimenet:  $\text{Vane} \in \mathbf{L}$

Előfeltétel:  $\text{Hossz}(\text{Tömb}) = N$

Utófeltétel:  $\text{Vane} \equiv \exists i \in [1..N] : \text{Páros}(\text{Tömb}_i)$

ahol

$\text{Páros}(x) = \text{igaz}$ , ha  $x \equiv 0 \pmod{2}$ ,

$\text{Páros}(x) = \text{hamis}$  minden egyéb esetben.

## Algoritmus:

**Eljárás**  $\text{Vane\_páros}(\text{Konstans } N: \text{Egész}; \text{Tömb}: \text{tTömb}; \text{Változó } \text{Vane}: \text{Logikai}):$

**Változó**

$I: \text{Egész}$

$I := 1$

**Ciklus amíg**  $I \leq N$  **és nem**  $\text{Páros}(\text{Tömb}(I))$

$I := I + 1$

**Ciklus vége**

$\text{Vane} = (I \leq N)$


**Eljárás vége**

# Érdekes programkészítési lépések

- Oktatásban ma elterjedt nyelvek:
  - (1) Pascal/Delphi, (2) (Visual) C++, C#, (3) Java, (4) Visual Basic
- Jövőben kaphatnak nagyobb szerepet:
  - (5) Ruby és (6) Python
- Kódolás folyamata
  - Kódolási szabályok
  - Kódolást segítő szolgáltatások
  - I/O
- Hibaüzenetek, fordítás utáni „rejtett” hibák
- Tesztelés, nyomkövető rendszer



# Programszerkesztés

- Angol nyelv ismerete
  - Pascal
    - Borland Pascal 7.0 (Free Pascal 2.2.0)
    - Borland Turbo Delphi Explorer (ingyenes)
- 

# Pascal nyelv jellemzői

- A programszerkezet könnyen követhető és memorizálható
- Alulról felfele építkeznek: a deklarációnak mindig meg kell előznie a használatot
- Értelmes alapszavak
  - Pl. I/O: `read`, `readln`, `write`, `writeln`
- Nem minden típusazonosító védett
  - `byte = string`
- Hozzáférési jog nélküli paraméterátadás



# Pascal nyelv jellemzői

- Következetlenség
  - `Procedure a(Var n: integer; b: byte);`
  - `a(n, b);`
  - Nem kell minden sor végén kitenni a pontosvesszőt (`if-then-else`)
- Összetett szerkezetek eleje-vége jelzése sem egységes
  - pl. `begin-end`, `record-end`, `while-end`, `repeat-until`
- „Csellengő else” probléma (C++, C#, Java nyelvekre is jellemző)
- Nem kisbetű-nagybetű érzékeny

```
if a=5 then
if a=4 then
else writeln(a);
```

# Borland Pascal 7.0

File Edit Search Run Compile Debug Tools Options Window Help

[■] PAROS.PAS 1=[↑]

```
Procedure Feldolgoz(N: Byte; Szamok: tTomb; Var Uane: Boolean);
```

```
Var  
  I: Byte;
```

```
Begin  
  I:=1;  
  While (I<=N) and ((Szamok[I] mod 2) <> 0) do  
  Begin  
    Inc(I);  
  End;  
  Uane:=(I <= N);  
End;
```

```
Procedure Ki(Uane: Boolean);  
Begin  
  If Uane Then Writeln('Talaltam paros szamot!')  
  Else Writeln('Nem talaltam paros szamot!');  
  Readkey;  
End;
```

49:38

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

# Delphi

```
40 procedure TParos.G_SzamokClick(Sender: TObject);  
  
·   Var  
·     I: Byte;  
·     Vane: Boolean;  
45  
·   begin  
·     I:=0;  
·     While (I<=(StrToInt(N.Text)-1)) and ((StrToInt(Szamok.Cells[I,0]) mod 2) <> 0) do  
·       Begin  
50         Inc(I);  
·       End;  
·     Vane :=(I<=(StrToInt(N.Text)-1));  
·     If Vane then ShowMessage('Találtam páros számot!')  
·       else ShowMessage('Nem találtam páros számot.');
```

52 Vane :=(I<=(StrToInt(N.Te)-1));  
· If Vane then ShowMessag property Text : TCaption;  
· else ShowMessa  
- end;  
·  
· end.

Text Property

Text - System.string

# Delphi – I/O

Páros szám

N:

Páros szám

1	9	33	23	<input type="text"/>
---	---	----	----	----------------------

N:

Paros\_prg

Találtam páros számot!

# C++

- Engedékenyebb, mint a Pascal
  - Változók deklarációja: ahol szükségét érezzük
- Tömörség
  - { }
  - operátorok
  - `i--`, `--i`, `i++`, `++i`: rontja az olvashatóságot
- I/O
  - `cin`, `cout` – nem utalnak egyértelműen a beolvasás és a kiíratás funkciójára
  - `cin >> n`, `cout << n`

# C++

- Hibaforrások
  - C-jellegű nyelvek összetett utasításainak törzse helyett véletlenül írt üres (;) utasítás
  - `switch` – `break` probléma (C#-ban már kötelező a `break`)
  - típushelyesség ellenőrzése közben a fordító csak figyelmeztet típusütközésnél, és automatikusan végzi el a konverziót – van, amikor helyesen, van amikor nem
- Kis- és nagybetűket megkülönbözteti

# Dev-C++

```
void be(int n, int szamok[])
{
    for (int i=0; i<n; ++i)
    {
        cout << "Kerem a(z) " << (i+1) << ". szamot: ";
        cin >> szamok[i];
    }
}

void feldolgoz(int n, int szamok[], bool& vane)
{
    int i = 0;
    while ((i < n) && ((szamok[i] % 2) != 0))
    {
        i++;
    }
    vane = (i < n);
}

void ki(bool vane)
{
    if (vane) cout << "Talaltam paros szamot!\n";
    else cout << "Nem talaltam paros szamot.\n";
}

int main(int argc, char *argv[])
{
    int n;
    cout << "Ez a program megmondja, hogy van-e paros szam a tombben.\n";
    cout << "Hany darab szam van a tombben?\n";
    cin >> n;
    int szamok[n];
    be(n,szamok);
    bool vane = false;
    feldolgoz(n,szamok,vane);
}
```

# Visual C++ 2008 Express

```
private: System::Void g_Keres_Click(System::Object^ sender, System::EventArgs^ e) {
    int i = 0;
    int n = System::Convert::ToInt32(N->Text);
    while ((i < n) && ((System::Convert::ToInt32(Szamok[i,0]->Value) % 2) != 0))
    {
        i++;
    }
    bool vane = (i < n);
    if (vane) MessageBox::Show("Találtam páros számot!");
    else MessageBox::Show("Nem találtam páros számot!");
}
```



# Visual C# 2008 Express

- Konzolos programok írása – hasonlóan a Delphihez
- Az osztály és objektum szintű metódusok tagkiválasztó operátora egységes (.)
- Világos paraméterkezelés
  - `int a`
  - `ref int a`
  - `out int a`
- Értelmes I/O parancsok (konzolos változat)
  - `Read`, `ReadLine`, `Write`, `WriteLine`
- A forrásfájl csak a kódot tartalmazza

# Visual C# 2008 Express

```
private void g_Keres_Click(object sender, EventArgs e)
{
    int i = 0;
    int n = System.Convert.ToInt32(N.Text);
    while ((i < n) && ((System.Convert.ToInt32(Szamok[i, 0].Value) % 2) != 0))
    {
        i++;
    }
    bool vane = (i < n);
    if (vane) MessageBox.Show("Találtam páros számot!");
    else MessageBox.Show("Nem találtam páros számot!");
}
```

# Java

```
1 import java.util.*;
2
3 public class Paros {
4
5     /**
6      * @param args
7      */
8     public static void main(String[] args) {
9         // TODO Auto-generated method stub
10        System.out.println("Ez a program eldönti, hogy van-e páros szám a sorozatban.");
11        int n = 0;
12        System.out.println("Add meg a tömb elemszámát!");
13        Scanner olvas = new Scanner(System.in);
14        n = olvas.nextInt();
15        int[] szamok = new int[n];
16        be(n, szamok);
17        boolean vane = feldolgoz(n, szamok);
18        ki(vane);
19    }
```

# Java - Eclipse

- Zárójelezés segítése

```
public static boolean feldolgoz(int n, int[] szamok)
{
    int i = 0;
    while ((i < n) && ((szamok[i] % 2) != 0))
    {
        i++;
    }
    return (i < n);
}
```

# Visual Basic 2008 Express

- A Basic egy implementációja
  - A strukturálatlan alapnyelv kibővítése strukturált és objektum orientált elemekkel
  - Nehéz továbblépés az elavult gyökerek miatt
- Támogatja a kezdőket – a nyelv és a környezet is

# Visual Basic 2008 Express

```
Private Sub g_keres_Click(ByVal sender As System.Object, ByVal e As  
    Dim i As Integer  
    i = 0  
    Dim n As Integer  
    n = Convert.ToInt32(Me.N.Text)  
    Szamok(n, 0).Value = 0  
    While i < n And Convert.ToInt32(Szamok(i, 0).Value) Mod 2 <> 0  
        i += 1  
    End While  
    Dim vane As Boolean = (i < n)  
    If vane Then  
        MessageBox.Show("Találtam páros számot")  
    Else  
        MessageBox.Show("Nem találtam páros számot")  
    End If  
End Sub
```

# Ruby (SciTE)

- Fiatal nyelv (1995.)
- A változó típusa értékadáskor dől el
  - Rontja az olvashatóságot és sok hibalehetőséget rejt
- A függvény nincs markánsan megkülönböztetve az eljárástól
- Ki- és beolvasás szintakszisa érthetetlen eltéréseket mutat
  - `gets` szöveg, `puts s`
  - `n = Integer(gets), puts n.to_s`

```
1 - def be(szamok)
2   puts 'Ez a program eldönti, hogy van-e páros szám a tömbben'
3   puts 'Írd be N értéket!'
4   n = 0
5   n = Integer(gets)
6 - for i in 1..n
7     puts 'Írd be a ' + i.to_s + ' . elemet:'
8     számok[i] = Integer(gets)
9   end
10  return n
11 end
12
13 - def feldolgoz(n, számok)
14   i = 1
15 - while (i <= n) and ((számok[i] % 2) != 0)
16     i = i + 1
17   end
18   return (i <= n)
19 end
20
21 - def ki(vane)
22 - if (vane) then puts 'Találtam páros számot'
23     else puts 'Nem találtam páros számot'
24   end
25 end
26 #Főprogram
27 számok = Array.new
28 n = be(szamok)
29 vane = feldolgoz(n, számok)
30 ki(vane)
```



# Python

- 1991-ben „született”
- Hasonlóság Rubyval
- Újdonság: margószabály
  - „Rákényszerít” az olvasható kódolásra
- Számlálós ciklus szokatlan szintakszis és nem kitalálható szemantika
  - `for i in range (1, n)`  
`for i in range (0, n)`

```
# -*- coding: cp1250 -*-
```

```
def be(szamok):  
    n = input('Add meg N értékét: ')  
    for i in range (0, n):  
        szamok.append(input('Add meg a tomb '+str(i+1)+' . értékét! '))  
    return n
```

```
def feldolgoz(n, szamok):  
    i = 0  
    while i < n and (szamok[i] % 2) != 0:  
        i = i + 1  
    return (i < n)
```

```
def ki(vane):  
    if vane:  
        print "Találtam páros számot!"  
    else:  
        print "Nem találtam páros számot"
```

```
# Főprogram
```

```
print 'Üdvözöllek, ez a program eldönti, hogy van-e páros szám a tömbben'
```

```
szamok = []
```

```
n = be(szamok)
```

```
vane = feldolgoz(n,szamok)
```

```
ki(vane)
```

# Fordítás, statikus szemantikai hibák

- Segít a hibaüzenet, ha tudatja
  - milyen hiba történt
  - hol és
  - milyen módon lehet megoldani azt.
- Két típushibát vizsgálunk
  - Utasítás/változó nevének elgépelése
  - Nem lezárt programblokk

# Borland Pascal

- Az első hibánál leáll

```
File Edit Search Run Compile Debug Tools Options Window Help
[ ]===== PAROS.PAS =====1=[↑]
Error 3: Unknown identifier.
While (I<=N) and ((Szamok[I] mod 2) <> 0) do
Begin
  Inc(I);
End;
Uane:=(I <= N);
End;

Procedure Ki(Uane: Boolean);
Begin
  If Uane Then Begin Writeln('Talaltam paros szamot!')
  Else Writeln('Nem talaltam paros szamot!');
  Readkey;
End;

Begin
  ClrScr;
  Be(N, Szamok);
  Feldolgoz(N, Szamok, Uane);
  Ki(Uane);
End.
*==== 48:22 =====
F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu
```

# Borland Pascal

File Edit Search Run Compile Debug Tools Options Window Help

[■] PAROS.PAS 1=[↑]

**Error 85: ";" expected.**

```
While (I<=N) and ((Szamok[I] mod 2) <> 0) do
Begin
  Inc(I);
End;
Uane:=(I <= N);
End;

Procedure Ki(Uane: Boolean);
Begin
  If Uane Then Begin Writeln('Talaltam paros szamot!')
  Else Writeln('Nem talaltam paros szamot!');
  Readkey;
End;

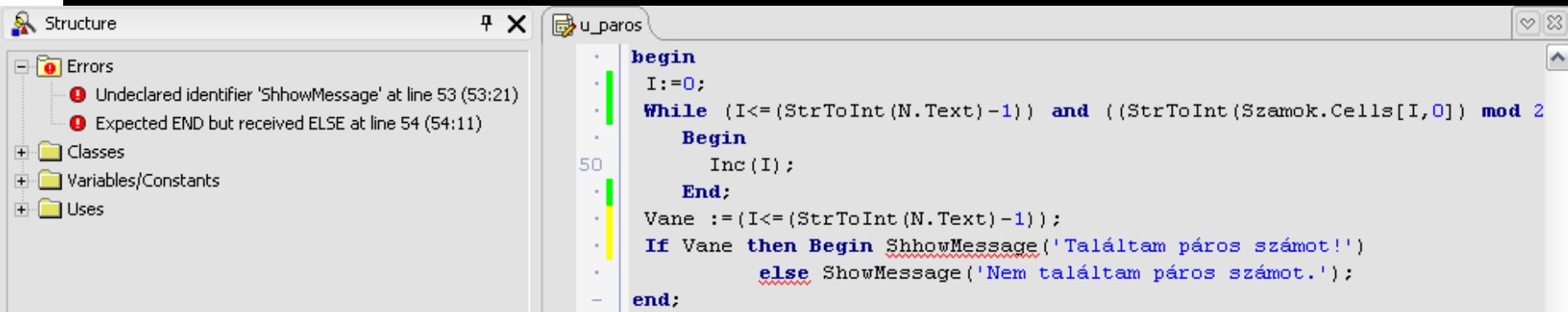
Begin
  ClrScr;
  Be(N, Szamok);
  Feldolgoz(N, Szamok, Uane);
  Ki(Uane);
End.
```

\* 49:11

F1 Help F2 Save F3 Open Alt+F9 Compile F9 Make Alt+F10 Local menu

# Turbo Delphi Explorer

- Lehetséges összes hibát kiírja
  - Kódolási időben is jelez hibákat!
- Tömör hibaüzenetek, rávilágítanak a problémára, a súgó is hasznos segítség



The screenshot shows the Turbo Delphi Explorer interface. On the left, the 'Structure' pane displays a tree view with 'Errors' expanded, showing two error messages: 'Undeclared identifier 'ShowMessage' at line 53 (53:21)' and 'Expected END but received ELSE at line 54 (54:11)'. Below this are folders for 'Classes', 'Variables/Constants', and 'Uses'. The main editor window, titled 'u\_páros', shows a Pascal code snippet with a line number 50 highlighted. The code is as follows:

```
begin
  I:=0;
  While (I<=(StrToInt(N.Text)-1)) and ((StrToInt(Szamok.Cells[I,0]) mod 2
  Begin
    Inc(I);
  End;
  Vane := (I<=(StrToInt(N.Text)-1));
  If Vane then Begin ShowMessage('Találtam páros számot!')
  else ShowMessage('Nem találtam páros számot.');
```

# Dev C++ - Visual C++ 2008 Express

- Kezdők számára nem egyértelmű hibaüzenetek

The screenshot shows the Visual C++ 2008 Express IDE. The top-left pane shows a project named 'Paros' with a file 'main.cpp'. The top-right pane shows the code for the 'ki' function:

```
void ki(bool vane)
{
    if (vane) { out << "Talaltam paros szamot!\n";
    else cout << "Nem talaltam paros szamot.\n";
}
```

The bottom pane shows the 'Fordító üzenetei' (Compiler Messages) window with the following table of errors:

Sor	Fájl	Üzenet
	C:\Dev-Cpp\main.cpp	In function `void ki(bool)':
32	C:\Dev-Cpp\main.cpp	`out' undeclared (first use this function) (Each undeclared identifier is reported only once for each function it appears in.)
33	C:\Dev-Cpp\main.cpp	expected primary-expression before "else"
33	C:\Dev-Cpp\main.cpp	expected ';' before "else"
37	C:\Dev-Cpp\main.cpp	a function-definition is not allowed here before '{' token
37	C:\Dev-Cpp\main.cpp	expected ';' or ':' before '{' token
46	C:\Dev-Cpp\main.cpp	expected '}' at end of input
	C:\Dev-Cpp\Makefile.win	[Build Error] [main.o] Error 1

# Visual C# 2008 Express

- Egyértelműbb, barátságosabb hibaüzenetek
- Kódolási időben, a háttérben is dolgozik a fordító
- Az ciklusváltozónak kezdőértéket kell adni (a példában `i`)



# Visual C# 2008 Express

```
private void g_Keres_Click(object sender, EventArgs e)
{
    int i = 0;
    int n = System.Convert.ToInt32(N.Text);
    while ((i < n) && ((System.Convert.ToInt32(Szamok[i, 0].Value) % 2) != 0))
    {
        i++;
    }
    bool vane = (i < n);
    if (vane) { MessageBox.Show("Találtam páros számot!");
    else MessageBox.Show("Nem találtam páros számot!");
}
```

## Error List

3 Errors 0 Warnings 0 Messages

	Description	File	Line	Column	Project
1	} expected	Paros.cs	41	2	Paros_cs
2	Invalid expression term 'else'	Paros.cs	38	18	Paros_cs
3	; expected	Paros.cs	38	23	Paros_cs



# Visual Basic 2008 Express

- Kész akarva nehéz ilyen hibákat generálni az intelligencia miatt
- 




```

Private Sub g_keres_Click(ByVal sender As System.Object, ByVal e As EventArgs)
    Dim i As Integer
    i = 0
    Dim n As Integer
    n = Convert.ToInt32(Me.N.Text)
    Szamok(n, 0).Value = 0
    While i < n And Convert.ToInt32(Szamok(i, 0).Value) Mod 2 <> 0
        i += 1
    End While
    Dim vane As Boolean = (i < n)
    If vane Then
        MeessageBox.Show("Találtam páros számot")
    End If
    Else
        MessageBox.Show("Nem találtam páros számot")
    End Else
End Sub

```

#### Error List

 3 Errors
  0 Warnings
  0 Messages

	Description	File	Line
	1 Name 'MeessageBox' is not declared.	Form1.vb	24
	2 'Else' must be preceded by a matching 'If' or 'ElseIf'.	Form1.vb	26
	3 'End' statement not valid.	Form1.vb	28

# Java - Eclipse

- A fordító a háttérben leleplezi a hibákat
- Jól láthatóan jelöli a hibás sorokat
- Érthetőek és lényegre törőek a hibaüzenetek

```
public static void ki(boolean vane)
{
    if (vane) { System.out.println("Találtam páros számot!");
    else System.out.println("Nem találtam páros számot!");
}
```

Problems Javadoc Declaration Search Console

3 errors, 0 warnings, 0 infos

Description	Resource	Path
Errors (3 items)		
Syntax error on token "else", delete this token	Paros.java	Páros
Syntax error, insert ")" to complete ClassBody	Paros.java	Páros
The method println(String) is undefined for the type PrintStream	Paros.java	Páros

```
36 public static boolean feldolgoz(int n, int[] szamok)
37 {
38     int i; // = 0;
39     while ((i < n) && ((szamok[i] % 2) != 0))
40     {
41         The local variable i may not have been initialized
42     }
43     return (i < n);
44 }
```



# Python

- Scriptnyelv – szintaktikai és szemantikai hibák futási időben derülnek ki
- Helyzetenként változik a hibaüzenet milyen pontos és specifikus

```
def ki(vane):  
    if vane  
    println "Találtam páros számot!"  
else:  
    print "Nem találtam páros számot"
```



```
def ki(vane):  
    if vane:  
    println "Találtam páros számot!"  
else:  
    print "Nem találtam páros számot"
```



```
def ki(vane):  
    if vane:  
        println "Találtam páros számot!"  
    else:  
        print "Nem találtam páros számot"
```

# Ruby - SciTE

- Még nehezebb értelmezni a hibaüzeneteket

```
21  - def ki(vane)
22  -   if (vane) then begin pputs 'Talaltam paros szamot'
23  -     | else puts 'Nem talaltam paros szamot'
24  -     end
25  -   end
26  -   #Főprogram
27  -   szamok = Array.new
28  -   n = be(szamok)
29  -   vane = feldolgoz(n, szamok)
30  -   ki(vane)
```

```
paros.rb:24: warning: else without rescue is useless
paros.rb:30: syntax error, unexpected $end, expecting kEND
  ki(vane)
  ^
```



# Tesztelés, hibakeresés

- Dinamikus szemantikai hibák sokkal kényelmetlenebbek → hibásan fut a program
- Előnyös tulajdonságok
  - Egy helyen találhatóak meg a funkciók
  - Minél több információ egy helyen
  - Lépésenkénti végrehajtás

# Borland Pascal

File Edit Search Run Compile Debug Tools Options Window Help

PAROS.PAS

1

```
Var
  I: Byte;

Begin
  <I:=1;>
  While <I<=N> or <<Szamok[I] mod 2> <> 0> do
  Begin
    Inc(I);
  End;
  Uane:=(I <= N);
End;

Procedure Ki(Uane: Boolean);
```

[■]

Watches

2=[↑]

```
Uane: False
N: 5
I: 4
Szamok: <1,3,5,6,7,0,0,0,0,0,0,0,0,0,0,0,0>
Szamok[I]: 6
```

F1 Help F7 Trace F8 Step ← Edit Ins Add Del Delete Alt+F10 Local menu

# Turbo Delphi Explorer

The screenshot displays the Turbo Delphi Explorer interface with the following components:

- Call Stack:** Shows the current call stack with the following entries:
  - u\_paros.TParos.G\_SzamokClick(???)
  - :0043da42 TControl.Click + \$6A
  - :00441282 TWinControl.WndProc + \$49E
  - :0042abfd TButtonControl.WndProc + \$71
  - :004413b8 DoControlMsg + \$28
  - :00441282 TWinControl.WndProc + \$49E
  - :00451152 TCustomForm.WndProc + \$4C6
  - :00440a0b TWinControl.MainWndProc + \$2F
- Watch List:** Contains the following watches:

Watch Name	Value
<input checked="" type="checkbox"/> I	16
<input checked="" type="checkbox"/> Vane	E2171 Variable 'Vane'...
<input checked="" type="checkbox"/> Szamok.Cells[I,0]	"
- Local Variables:** Shows the local variables for the current procedure:

Name	Value
Self	[[csInheritable], False, (0, ...
Sender	E2171 Variable 'Sender' ina...
I	16
Vane	E2171 Variable 'Vane' inacc...
- Code Editor:** Displays the source code for the procedure `TParos.G_SzamokClick`. The current line is 48, which is highlighted in blue. The code is as follows:

```
48 While (I <= (StrToInt(N.Text) - 1)) or ((StrToInt(Szamok.Cells[I,  
50 Begin N.Text | '5'  
Inc (I);  
End;  
Vane := (I <= (StrToInt(N.Text) - 1));  
If Vane then ShowMessage('Találtam páros számot!')  
else ShowMessage('Nem találtam páros számot.');
```

# Dev C++, Eclipse

- Külön hibakereső mód
  - Dev C++: a fordító beállításain kell változtatni → bonyolult
  - Eclipse: egy kattintás az egész!

Debug

Paros [Java Application]

- Paros at localhost:3617
  - Thread [main] (Suspended (breakpoint at line 39 in Paros))
    - Paros.feldolgoz(int, int[]) line: 39
    - Paros.main(String[]) line: 17
- C:\Program Files\Java\jre1.6.0\_03\bin\javaw.exe (2008.02.26. 11)

Variables

Name	Value
n	5
szamok	int[5] (id=16)
i	0

[1, 3, 5, 7, 9]

```
36 public static boolean feldolgoz(int n, int[] számok)
37 {
38     int i = 0;
39     while ((i < n) || ((számok[i] % 2) != 0))
40     {
41         i++;
42     }
43     return (i < n);
44 }
45
```

Outline

- import declarations
- Paros
  - main(String[])
  - be(int, int[])
  - feldolgoz(int, int[])
  - ki(boolean)

Console

Paros [Java Application] C:\Program Files\Java\jre1.6.0\_03\bin\javaw.exe (2008.02.26. 16:49:51)

```
5
Írd be a 4. elemet!
7
Írd be a 5. elemet!
9
```

# Visual C# 2008 Express

```
private void g_Keres_Click(object sender, EventArgs e)
{
    int i = 0;
    int n = System.Convert.ToInt32(N.Text);
    while ((i < n) || ((System.Convert.ToInt32(Szamok[i, 0].Value) % 2) != 0))
    {
        i++;
    }
    bool vane = (i < n);
    if (vane) MessageBox.Show("Vannak párosok.");
    else MessageBox.Show("Nincsenek párosok.");
}
```

## ArgumentOutOfRangeException was unhandled

Az index az engedélyezett tartományon kívül esik. Az index nem lehet negatív, és kisebbnek kell lennie a gyűjtemény méreténél.  
Paraméter neve: index

### Troubleshooting tips:

[Make sure the arguments to this method have valid values.](#)

[If you are working with a collection, make sure the index is less than the size of the collection.](#)

[When using the overloaded two-argument FindString or FindExactString methods with a ComboBox or ListBox, check the startIndex parameter.](#)

[Get general help for this exception.](#)

[Search for more Help Online...](#)

### Actions:

[View Detail...](#)


[Copy exception detail to the clipboard](#)

Name	Value
this	{WindowsFormsApplication1}
sender	{Text = "Keres"}
e	{X = 23 Y = 10 Button = L}
i	5
n	5
vane	false

int  
bool



# Python, Ruby

- A tárgyalt környezetek hátránya
    - Nem rendelkeznek nyomkövető rendszerrel
    - A programozónak kell kiíratnia a változók értékeit
- 

```
13  - def feldolgoz(n, szamok)
14      i = 1
15      #Nyomkövetés
16      puts 'n: ' + n.to_s
17      puts 'Belepünk a ciklusba...'
18  - while (i <= n) or ((szamok[i] % 2) != 0)
19      |   puts 'i: ' + i.to_s
20      |   puts 'szamok[i]: ' + szamok[i].to_s
21      |   puts '(i <= n): ' + (i <= n).to_s
22      |   #Nyomkövetés vége
23      |   i = i + 1
24      |   end
25      |   #Nyomkövetés
26      |   puts 'Kileptünk a ciklusból'
27      |   puts 'i: ' + i.to_s
28      |   puts '(i <= n): ' + (i <= n).to_s
29      |   #Nyomkövetés vége
30      |   return (i <= n)
31  end
```



```
n: 5
Belepunk a ciklusba...
i: 1
szamok[i]: 1
(i <= n): true
i: 2
szamok[i]: 3
(i <= n): true
i: 3
szamok[i]: 5
(i <= n): true
i: 4
szamok[i]: 7
(i <= n): true
i: 5
szamok[i]: 9
(i <= n): true
paros.rb:18:in `feldolgoz': undefined method `%` for nil:NilClass (NoMethodError)
)
    from paros.rb:41
```

# Összefoglalás – nyelvek

- Nincs tökéletes nyelv és környezet
- Oktatási szempontok
  - Értelmes alapszavak
  - Milyen könnyű megírni az első programot?
  - Mennyire áll közel az algoritmikus nyelvünkhöz?
  - A tanuló megérti a programozás alapvető elemeinek helyét, szerepét és használatát

# Összefoglalás – nyelvek

- Pascal
  - Erős és szigorú típusosság
  - Átlátható és memorizálható programszerkezet
  - Jó alap a továbblépéshez
    - Objektum orientáltság (OO)
    - 4GL környezet
  - → Jó első imperatív nyelv
- C++, C#, Java
  - OO új távlatok nyelvi szinten (Pascalhoz képest)
  - C++ túl bonyolult első imperatív nyelvként
  - C#, Java: letisztultabb, kiforrottabb → jobb választás továbblépés esetén

# Összefoglalás – nyelvek

- Visual Basic
  - Rengeteg előnyös tulajdonság kezdők számára
  - A nyelvi gyökerek és az erős kötődés a fejlesztői környezethez megnehezíti a továbblépést
- Ruby, Python
  - Gyenge típusosság
    - Nem támogatják a haladó programozási stílus kialakulását
    - Rontja a kód olvashatóságát

# Összefoglalás – környezetek

- Olvashatóság
  - Nincsenek lényeges különbségek
  - Intellisence
- Hibaüzenetek
  - Delphi, Visual C# és Basic 2008 Express, Exlipse 😊
  - (Visual) C++, Python, Ruby ☹️
- Nyomkövetés
  - Nincsenek jelentős különbségek

# Összefoglalás – paradigmák

- Érdeemes OO nyelvet/környezetet használni?
  - Be lehet vezetni egyszerű módon
    - Algoritmikus nyelv alkalmassá tétele OO-ra
  - Hogyan hat ez az oktatás hatékonyságára?
- Kezdeti szakasz: Programozási tételek, algoritmizálás
  - OO nélkül, algoritmus szempontjából fontos nyelvi elemek
  - Nem hasznos egy 4GL rendszer vagy a tárgyalt scriptnyelvek használata



**KÖSZÖNÖM A FIGYELMET!**