

# Programovací jazyky

# Opakování – test (1)

---

- Napište definici algoritmu:
- Napište alespoň pět vlastností algoritmu:

# Opakování – test (2)

---

- Napište definici algoritmu:
  - Algoritmus je přesný postup, kterým lze vyřešit daný typ úlohy.

# Opakování – test (3)

---

➤ **Napište alespoň pět vlastností algoritmu:**

- Konečnost
- Determinismus
- Vstup
- Výstup
- Efektivita
- Obecnost
- Rezultativnost

# Programovací jazyk

---

- **Programovací jazyk** je prostředek pro zápis algoritmu, jež mohou být provedeny na počítači.
- **Počítačový program** je zápis algoritmu ve zvoleném programovacím jazyce se nazývá

# Formální jazyk

---

- **Programovací jazyk** je soubor pravidel pro zápis algoritmu - je to tzv. formální jazyk
- **Formální jazyk** - množina konečných slov (tj. slov konečné délky) nad určitou abecedou.

# Důležité pojmy (1)

---

## ➤ Zdrojový kód (kód, "zdroják")

- série sdělení psaných na počítači v nějakém člověku čitelném programovacím jazyce

- příklad:

```
for (i=0; i<sloup; i++){  
    for (j=0; j<rad; j++) {  
        pole[i][j]=' ';}}}
```

# Důležité pojmy (2)

---

## ➤ **Strojový kód**

- je posloupnost instrukcí procesoru vyjádřená čísly.
- příklad:

:1000000075B0AA7590AA12001575B05575905  
51265



# Důležité pojmy (3)

---

## ➤ Překladač

- stroj respektive program, provádějící překlad z nějakého vstupního jazyka do jazyka výstupního. Nejčastěji bývá tento pojem používán v programování, kde označuje nástroje překládající algoritmus zapsaný ve vyšším programovacím jazyce do jazyka strojového, nebo spíše do strojového kódu.

# Důležité pojmy (4)

---

## ➤ **Překladač – jednotky:**

- Lexikální analyzátor
- Syntaktický analyzátor
- Sémantický analyzátor
- Překladač do mezikódu
- Optimalizátor
- Generátor kódu

# Důležité pojmy (5)

---

## ➤ **Emulátor**

- druh software umožňující běh počítačových programů na jiné platformě (pod jiným operačním systémem, jiné architektuře), než pro kterou byly původně vyrobeny

- **Příklad:**

Možnost spustit hru vytvořenou pro Linux pod Microsoft Windows

# Důležité pojmy (5)

---

## ➤ Interpret

- speciální software umožňující spouštět programy bez nutnosti kompilace
- překládat jej počítači přímo za běhu programu
- **Výhody:**
  - jednoduchá údržba kódu
  - nemusí se provádět kompilace
- **Nevýhody:**
  - nižší rychlost
  - chyby odhalované až při běhu programu

# Typy programovacích jazyků (1)

---

## ➤ Dle míry abstrakce:

- vyšší programovací jazyky (většina jazyků)
- nižší programovací jazyky (např. assembler)

Assembler – jazyk symbolických adres

```
mov    ah,1
int    16h
jz     pok
xor    ah,ah
int    16h
```

# Typy programovacích jazyků (2)

```

D:\PETR\DOCUME~1\PETRHL~1\DOKUME~1\AS51\AS51.EXE
Load Edit Compile Debug Option BreakPoint View Modify Quit
VIEW TEXT
... \DOKUME~1\AS51\TEST
CPU
00,000 000.0 PC 0000
Acc 00 00000000 DPTR 0000
PSW 00 00000000
B 00 00000000
SP 07 00000111
Timer
ICON 00
TMOD 00
Int BREAK
No Break
IE 00
IP 00
Hi Lo
T0 00 00
T1 00 00
SCON
PCON 00
RBUF 00
TBUF 00
P0 P1 P2 P3
FF FF FF FF
R0-R1-R2-R3-R4-R5-R6-R7
RB0 00 00 00 00 00 00 00 00
Intern MEMORY
0000 00 00 00 00 00 00 00 00 .....
0008 00 00 00 00 00 00 00 00 .....
0010 00 00 00 00 00 00 00 00 .....
0018 00 00 00 00 00 00 00 00 .....
R0:00-R1:00
PSW : CY AC F0 RS1 RS0 OV xx P
0 0 0 0 0 0 0 0
F1:Help F3:Clear F4:Reset F5:Modify F6:Animate F7:Trace F8:Step F9:Run F10:Menu
  
```

# Typy programovacích jazyků (3)

---

## ➤ Dle způsobu překladač a spuštění:

- kompilované programovací jazyky (Pascal, C..)
- interpretované programovací jazyky (Basic, Perl, Python, shell )
  - Druhy:
    - pouze se interpretují
    - které se překládají do mezikódu
    - spuštění za běhu programu překládají do strojového kódu

# Typy programovacích jazyků (4)

---

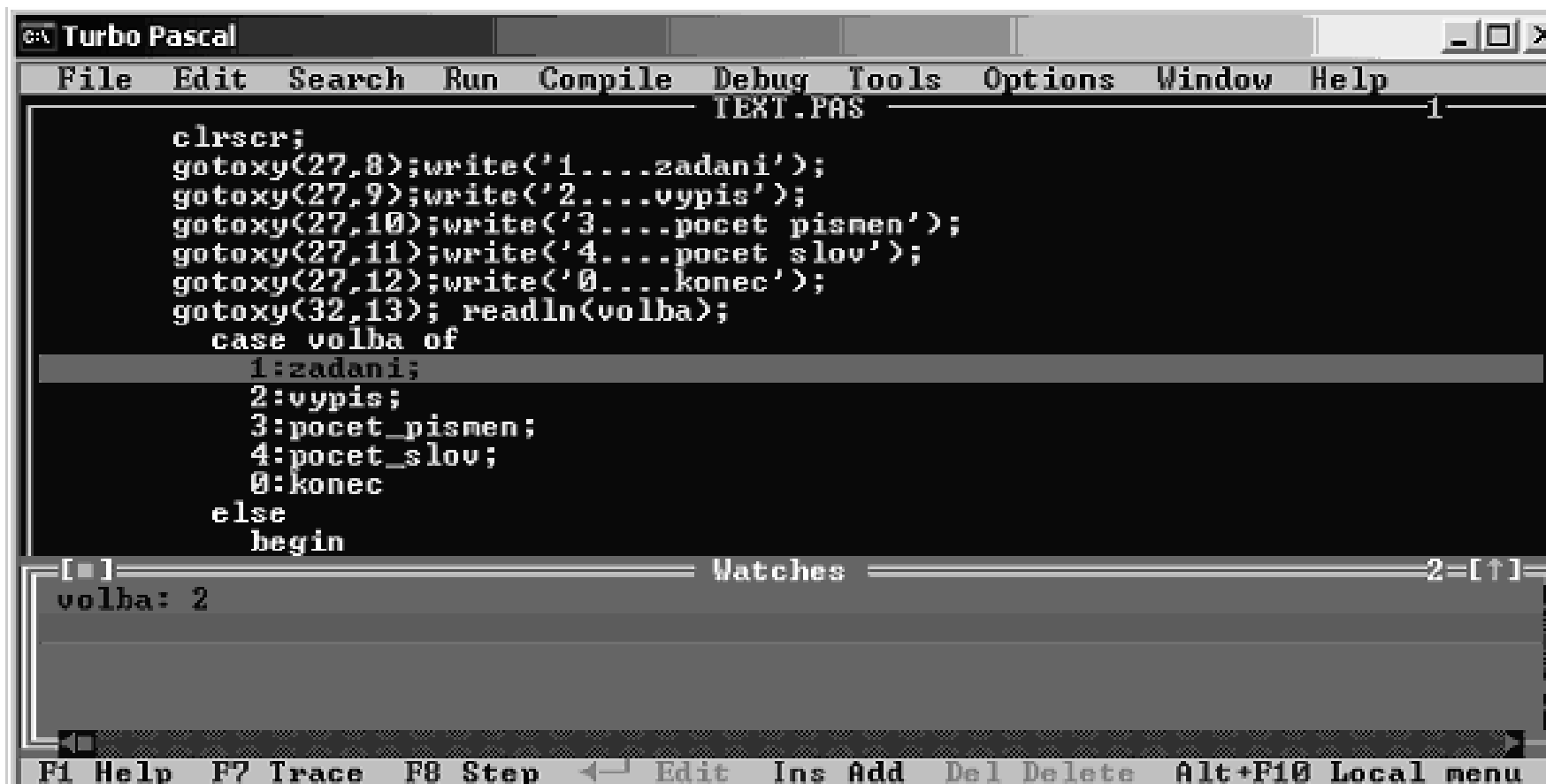
## ➤ Dělení vyšších programovacích jazyků:

- Procedurální (imperativní)
  - Strukturované (Pascal, C...)
  - Objektově orientované (C++, Java)
- Neprocedurální (deskriptivní)
  - Funkcionální (Lisp, Haskell)
  - Logické (Prolog)



# Příklad programovacích jazyků (1)

---



The screenshot shows the Turbo Pascal IDE with a window titled "Turbo Pascal" and a menu bar including File, Edit, Search, Run, Compile, Debug, Tools, Options, Window, and Help. The main editor displays the following Pascal code in a file named "TEXT.PAS":

```
clrscr;
gotoxy(27,8);write('1....zadani');
gotoxy(27,9);write('2....vypis');
gotoxy(27,10);write('3....pocet pismen');
gotoxy(27,11);write('4....pocet slov');
gotoxy(27,12);write('0....konec');
gotoxy(32,13); readln(volba);
  case volba of
    1:zadani;
    2:vypis;
    3:pocet_pismen;
    4:pocet_slov;
    0:konec
  else
  begin
```

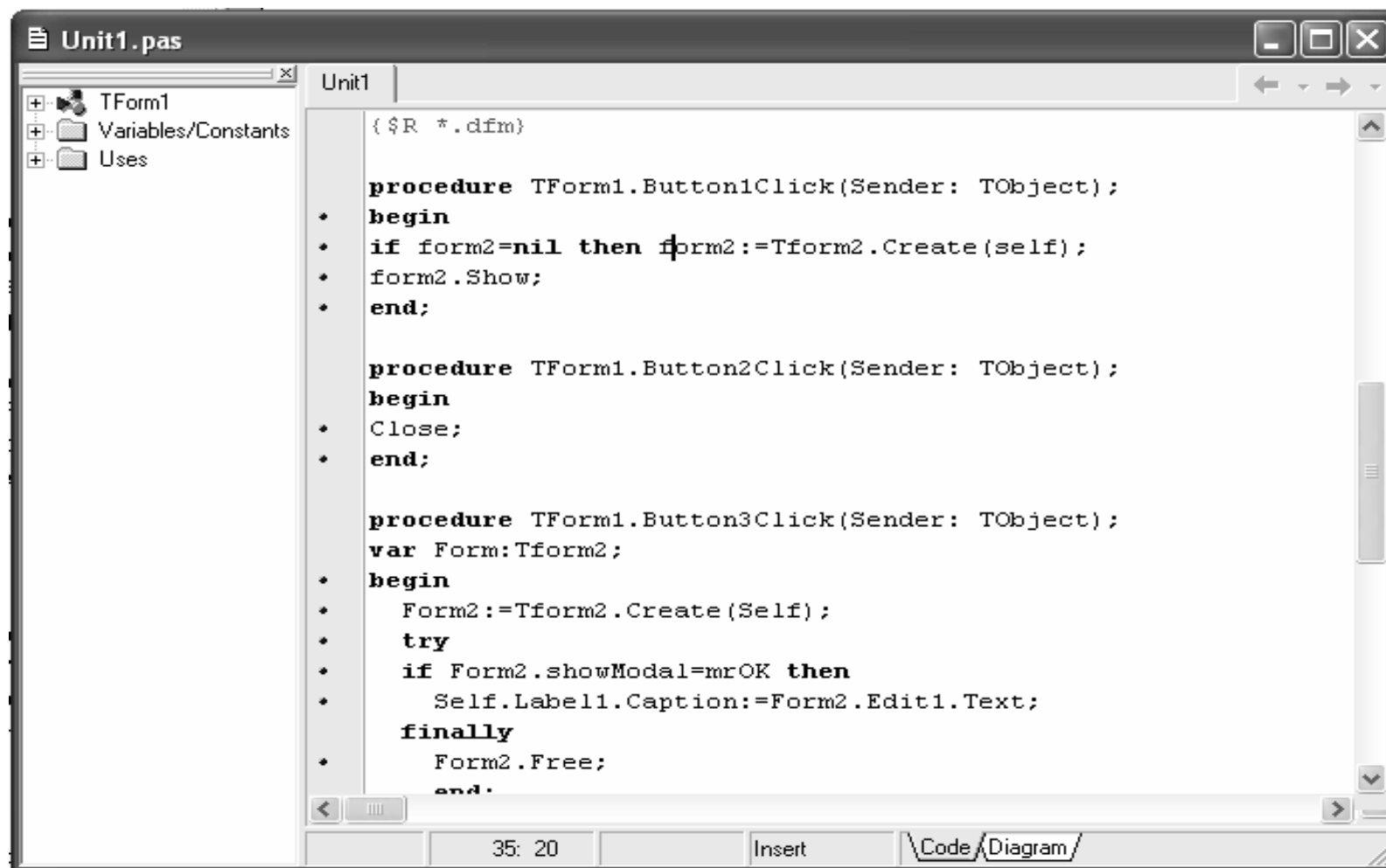
Below the editor is a "Watches" window showing the variable "volba" with the value "2". The bottom status bar contains the text: "F1 Help F7 Trace F8 Step ← Edit Ins Add Del Delete Alt+F10 Local menu".

# Příklad programovacích jazyků (2)

---



# Příklad programovacích jazyků (3)



```
Unit1

($R *.dfm)

procedure TForm1.Button1Click(Sender: TObject);
* begin
* if form2=nil then form2:=Tform2.Create(self);
* form2.Show;
* end;

procedure TForm1.Button2Click(Sender: TObject);
begin
* Close;
* end;

procedure TForm1.Button3Click(Sender: TObject);
var Form:Tform2;
* begin
*   Form2:=Tform2.Create(Self);
*   try
*   if Form2.showModal=mrOK then
*     Self.Label1.Caption:=Form2.Edit1.Text;
*   finally
*     Form2.Free;
*   end;
```

# Příklad programovacích jazyků (4)

---

- Fibonacciho funkce (Haskell):

$\text{fib}' :: \text{Int} \rightarrow \text{Integer}$

$\text{fib}' 0 = 0$

$\text{fib}' 1 = 1$   
 $\text{fib}' n = \text{fib}' (n-2) + \text{fib}' (n-1)$   
 $\text{fib} ::$

$\text{Int} \rightarrow \text{Integer}$   
 $\text{fib } n = f$

# „Životopis“ programu

---

- Zdrojový kód (např. Pascal)
- Makroexpanze
- Překlad (compilation)
- Sestavení (linking)
- Spuštění programu

# Programovací jazyk Pascal (1)

---

- Název zvolen na počest francouzského filozofa, matematika a fyzika Blaise Pascala
- prof. Niklaus Wirth, Curych 1971 - 1973
- **Základní cíle:**
  - vytvořit jazyk vhodný pro výuku programování
  - definovat jazyk, jehož implementace na současných počítačích vedou k efektivním a spolehlivým programům

# Programovací jazyk Pascal (2)

---

## ➤ **Hlavní výhody:**

- standardizace řídicích struktur, které vedou k vytváření dobře strukturovaných a čitelných programů
- bohatost datových typů
- relativně snadná přenositelnost na různé typy počítačů

# Programovací jazyk Pascal (3)

---

- 1981 - zveřejněn návrh normy ISO jazyka Pascal
- Borland (komerční úspěch):
  - Turbo (Borland) Pascal (pro MS-DOS)
  - Turbo Pascal for Windows
  - Borland Delphi (pro Windows 9x, Me, NT, 2000, XP)