

Imperatív programozás

Bevezetés



Kozsik Tamás és mások

ELTE Eötvös Loránd Tudományegyetem

A tárgy célja

- Fogalomrendszer
- Terminológia magyarul és angolul
- Tudatos nyelvhasználat
 - Imperatív programozás
 - Procedurális programozás
 - Moduláris programozás
- Részben: programozási készségek
- Linux és parancssori eszközök használata



Használt programozási nyelvek

- C
- Python



1 Paradigmák és nyelvek

- Alacsony szintű és magas szintű programozás
- Programozási nyelvek történelme

Programozási nyelvek

- Ember-gép kommunikáció
- Ember-ember kommunikáció



Programozási paradigmák

Gondolkodási sémák, szükséges nyelvi eszközök

Például:

- Imperatív programozás
- Funkcionális programozás
- Logikai programozás

- Procedurális programozás
- Moduláris programozás
- Objektumelvű programozás

- Szekvenciális programozás
- Konkurens programozás
- Párhuzamos programozás
- Elosztott programozás

- Aspektuselvű programozás
- Komponenselvű programozás
- Szolgáltatáselvű programozás
- Szerződésalapú programozás



Assembly

quickSort:

.LFB1:

```
.cfi_startproc
pushl    %ebp
.cfi_def_cfa_offset 8
.cfi_offset 5, -8
movl     %esp, %ebp
.cfi_def_cfa_register 5
subl     $40, %esp
movl     12(%ebp), %eax
cmpl     16(%ebp), %eax
jge      .L6
movl     16(%ebp), %eax
movl     %eax, 8(%esp)
movl     12(%ebp), %eax
movl     %eax, 4(%esp)
```



„Magas szintű” programozási nyelvek

- Fortran
- LISP
- Algol
- COBOL
- BASIC
- C

stb.



Modern, kényelmes nyelvek

- **Python**
- **Haskell**
- C++
- Java
- Ada

stb.



Ada Lovelace (Analytical Engine, Charles Babbage)

Diagram for the computation by the Engine of the Numbers of Bernoulli. See Note G. (page 722 *et seq.*)

Number of Operation.	Nature of Operation.	Variables acted upon.	Variables receiving results.	Indication of change in the value on any Variable.	Statement of Results.	Data.												Working Variables.												Result Variables.					
						$1V_1$	$1V_2$	$1V_3$	$1V_4$	$1V_5$	$1V_6$	$1V_7$	$1V_8$	$1V_9$	$1V_{10}$	$1V_{11}$	$1V_{12}$	$1V_{13}$	$1V_{14}$	$1V_{15}$	$1V_{16}$	$1V_{17}$	$1V_{18}$	$1V_{19}$	$1V_{20}$	$1V_{21}$	$1V_{22}$	$1V_{23}$	$1V_{24}$	$1V_{25}$	$1V_{26}$	$1V_{27}$	$1V_{28}$	$1V_{29}$	$1V_{30}$
						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
						1	2	n																											
1	\times	$1V_2 \times 1V_3$	$1V_2, 1V_3$	$1V_2 = 1V_2$	$= 2n$...	2	n	2n	2n	2n
2	$-$	$1V_4 - 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$...	1
3	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n+1$...	1
4	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$
5	$+$	$1V_{11} + 1V_5$	$2V_{11}$	$1V_5 = 1V_5$	$= 2n-1$...	2
6	$-$	$1V_{11} - 1V_{12}$	$2V_{11}$	$1V_{11} = 1V_{11}$	$= 2n-1$
7	$-$	$1V_5 - 1V_6$	$2V_5$	$1V_5 = 1V_5$	$= n-1 (=3)$...	1	...	n
8	$+$	$1V_5 + 1V_6$	$2V_5$	$1V_5 = 1V_5$	$= 2n-1$...	2
9	$+$	$1V_5 + 1V_6$	$2V_5$	$1V_5 = 1V_5$	$= 2n-1$
10	\times	$1V_{11} \times 1V_{12}$	$2V_{11}$	$1V_{11} = 1V_{11}$	$= 2n-1$
11	$+$	$1V_{11} + 1V_{12}$	$2V_{11}$	$1V_{11} = 1V_{11}$	$= 2n-1$
12	$-$	$1V_{11} - 1V_{12}$	$2V_{11}$	$1V_{11} = 1V_{11}$	$= 2n-1$...	1
13	$-$	$1V_4 - 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$...	1
14	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$...	1
15	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$
16	\times	$1V_4 \times 1V_{11}$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$
17	$-$	$1V_4 - 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$...	1
18	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$...	1
19	$+$	$1V_4 + 1V_5$	$2V_4$	$1V_4 = 1V_4$	$= 2n-1$
20	\times	$1V_{11} \times 1V_{12}$	$2V_{11}$	$1V_{11} = 1V_{11}$	$= 2n-1$															



Augusta Ada King, Countess of Lovelace (née Byron, 1815–1852)



A programozás őskora

- Fizikai huzalozás (pl. ENIAC, 1945)
- Gépi kód (Neumann-architektúra, 1945)
- Assembly (1949–)
- Magas szintű programozási nyelvek
 - Plankalkül (Konrad Zuse, 1942–1945)
 - Fortran (John Backus et al., 1954)
 - LISP (John McCarthy, 1958)
 - Algol (1958, 1960, 1968)
 - COBOL (1959)
 - BASIC (Kemény–Kurtz, 1964)



Néhány fontos nyelv

- Simula-67 (Dahl–Nygaard, 1967)
- Pascal (Niklaus Wirth, 1970)
- C (Dennis Ritchie, 1972)
- Ada (1980)
- SQL (Chamberlin–Boyce, 1974)
- C++ (Bjarne Stroustrup, 1985)
- Eiffel (Bertrand Meyer, 1986)
- Erlang (Armstrong–Virding–Williams, 1986)
- Haskell (1990)
- Python (Guido van Rossum, 1990)
- Java (James Gosling, 1995)
- JavaScript (Brendan Eich, 1995)
- PHP (Rasmus Lerdorf, 1995)
- C# (2000)
- Scala (Martin Odersky, 2004)

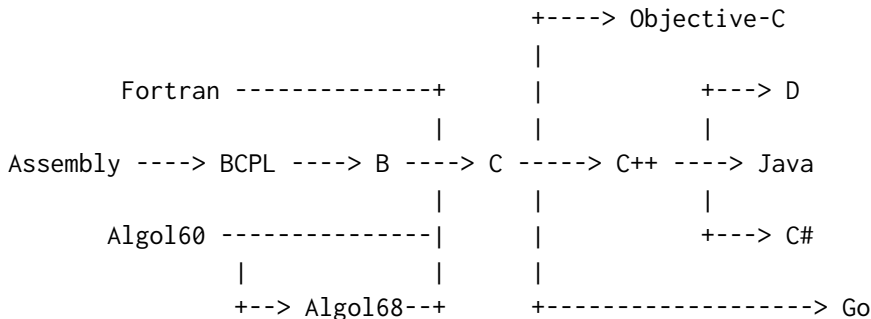


Legnépszerűbb nyelvek (2018. szeptember, TIOBE-index)

Sep 2018	Sep 2017	Change	Programming Language	Ratings	Change
1	1		Java	17.436%	+4.75%
2	2		C	15.447%	+8.06%
3	5	▲	Python	7.653%	+4.67%
4	3	▼	C++	7.394%	+1.83%
5	8	▲	Visual Basic .NET	5.308%	+3.33%
6	4	▼	C#	3.295%	-1.48%
7	6	▼	PHP	2.775%	+0.57%
8	7	▼	JavaScript	2.131%	+0.11%
9	-	▲▲	SQL	2.062%	+2.06%
10	18	▲▲	Objective-C	1.509%	+0.00%
11	12	▲	Delphi/Object Pascal	1.292%	-0.49%
12	10	▼	Ruby	1.291%	-0.64%
13	16	▲	MATLAB	1.276%	-0.35%
14	15	▲	Assembly language	1.232%	-0.41%
15	13	▼	Swift	1.223%	-0.54%
16	17	▲	Go	1.081%	-0.49%
17	9	▼▼	Perl	1.073%	-0.88%
18	11	▼▼	R	1.016%	-0.80%



A C nyelv kialakulása



A C nyelv fejlődése

- 1969 Ken Thompson kifejleszti a B nyelvet (egy egyszerűsített BCPL)
- 1969 Ken Thompson, Dennis Ritchie és mások elkezdnek dolgozni a UNIX-on
- 1972 Dennis Ritchie kifejleszti a C nyelvet
- 1972-73 UNIX kernel-t újraírják C-ben
- 1977 Johnson Portable C Compiler-e
- 1978 Brian Kernighan és Dennis Ritchie: The C Programming Language könyve
- 1989 ANSI C standard (C90) (32 kulcsszó)
- 1999 ANSI C99 standard (+5 kulcsszó)
- 2011 ANSI C11 standard (+7 kulcsszó)
- 2018 C18 ISO/IEC standard

Mi alapvetően az ANSI C-t, azaz a C90-et fogjuk használni.



A Python nyelv kialakulása és fejlődése

- Inspiráció: ABC, ALGOL 68, APL C, C++, CLU, Dylan, Haskell, Icon, Java, Lisp, Modula-3, Perl, Standard ML
- 1990 Guido van Rossum
- 2000 Python 2.0
- 2008 Python 3.0

Mi a python3-at fogjuk használni, az iparban nagyon sok a python2.7.

