

A programozási nyelvek elemei (Lexikális elemek)

- A programok fordítási egységei lexikális elemek sorozatai.
- A lexikális elemek karaktersorozatok, határoló jelekkel elválasztva.
- A nyelvekben általában megkülönböztetjük a grafikus karaktereket (a Latin-1 vagy az angol abc betűi, számjegyek és speciális karakterek - pl. ' ' # & ' () , * - + / : ; < = _ [] { } | . - -), a formátum vezérlő karaktereket (az ISO 6429 szabványban: character tabulation (HT), line tabulation (VT), carriage return (CR), line feed (LF), and form feed (FF)), és az egyéb (implementáció függő) vezérlő karaktereket.

2005.02.24.

1

A lexikális elem

- határoló,
- azonosító
- numerikus literál,
- karakter literál,
- string literál,
- megjegyzés

2005.02.24.

2

Milyen karakterek használhatók a nyelvben?

- **Pascal (ISO 7185:1990):**
 - csak minimum követelmények, pl.: '0'..'9' rendezett és folytonos kell legyen, 'A'..'Z' rendezett, de nem feltétlenül folytonos, 'a'..'z' nem kötelező, de ha van, akkor mint a nagybetűk.
 - Nincs előírva a lehetséges karakterek halmaza, különböző implementációk különböző kódolást használhatnak(!) (pl. EBCDIC vagy ASCII) => különböző implementációk másképpen viselkedhetnek (' ' ∈ 'A'..'Z' EBCDIC-ben és ' ' ∉ 'A'..'Z' ASCII-ben, vagy 'A' < 'I' EBCDIC-ben és 'I' < 'A' ASCII-ben).
 - => ez nem tesz jót a programok hordozhatóságának

2005.02.24.

3

• C++(ISO/IEC 14882:1998):

- Az alap karakterkészlet 96 elemű: a space, a horizontal tab, vertical tab, new-line, form feed vezérlő karakterei, és a következő 91 grafikus karakter: 'a'..'z' 'A'..'Z' '0'..'9' " # & ' () , * - + / : ; < > = _ [] { } | . % ? ^ ~ ! \
- ISO 10646 szerint kódolva
- => ez precízebb, mint az 1990-es, ahol nem volt a kód előírva.

2005.02.24.

4

- **CLU, Eiffel:**
 - ASCII kód a megengedett.
- **ADA95 (ISO/IEC 8652:1995) :**
 - ISO 10646 BMP; Row00: basic Latin és Latin-1 Supplement
- **Java:**
 - "Java programs are written using the Unicode character set, version 2.0"
- **C#:**
 - „A source file is an ordered sequence of Unicode characters.”

2005.02.24.

5

Melyek az elhatároló jelek?

- A legtöbb programozási nyelvben a helyköz (space), a tab, a sorvége elhatároló jel.
- Van sok egy karakterből álló elhatároló jel, mint pl. :
Pascal: & ' () + - * / : . ; < = >
C++: & % ^ ' () + - { } | ~ [] \ " * / : . ; , < = > ! ?
Java: () { } [] ; , . + - * / : < = > ! ~ ? : & | ^ %
ADA95: & ' () * + , - . / : ; < = > | stb.

2005.02.24.

6

Melyek az elhatároló jelek?

- Van számos több karakterből álló elhatároló jel is, mint pl. :

Pascal: := >= <= <> << ** ..

C++: -> ++ -- * -> * == <= >= && || << >> <=> >>= != .. += -= &= ^= |= :: *= /=

Java: == <= >= << != && || ++ -- >> >>> += -= *= /= &= ^= %= <<= >>= >>>=

ADA95: => .. ** := /= >= <= << >> <> -- stb.

2005.02.24.

7

Megkülönbözteti-e a nyelv a kis- és nagybetűket?

- **Pascal, Fortran, CLU, ADA...**: ekvivalensek
- **C, C++, Modula, Perl, Java, C#**: minden betű különböző
- **Eiffel**: a kis- és a nagybetűk ekvivalensek, de használhatjuk ugyanazt a nevet egy objektumra és a típusára
 - pl. : a: A megengedett

2005.02.24.

8

Azonosítók

- Milyen karakterek használhatóak az azonosítók leírására?
- Mi az azonosító megengedett szintaxisa?
- Van-e hosszúsági megkötés az azonosítókra?
- Vannak-e kötött szavak?
Van-e különbség a kulcsszavak és az előre definiált szavak között?

2005.02.24.

9

Milyen karakterek használhatóak?

- **Pascal** (ISO 7185:1990):
 - betűk ('A'..'Z', 'a'..'z') és számjegyek ('0'..'9')
- **C++** (ISO/IEC 14882:1998):
 - betűk ('A'..'Z', 'a'..'z'), számjegyek ('0'..'9') és '_'
- **ADA95** (ISO/IEC 8652:1995):
 - az ISO 10646 BMP Row 00 bármely karaktere, amelynek a neve "Latin Capital Letter" és "Latin Small Letter" (ezek az ún. **identifier_letters**), számjegyek ('0'..'9') és '_'

2005.02.24.

10

Milyen karakterek használhatóak?

- **Java**:
 - A Unicode 2.0 betűi és számjegyei, a '\$' és '_'
- **CLU, Eiffel**:
 - betűk ('A'..'Z', 'a'..'z'), számjegyek ('0'..'9') és '_'
- **Perl**:
 - az előzőeken túl \$, @ és % jellel is kezdődhetnek
\$ a skalárokat, @ számmal indexelt tömböt, % asszociatív tömböt jelöl.

2005.02.24.

11

Milyen karakterek használhatóak?

- **C#**:
Az azonosítókat a Unicode 3.0 szabvány ajánlásának megfelelően kell írni.
Érdekeség, hogy a kulcsszavak a @ bevezető jellel használhatóak azonosítóként is (például: @bool). Ez a jel azonosítót nem vezethet be.
Ezt a lehetőséget a programok hordozhatósága illetve "átjárhatósága" miatt vezették be. A @ prefixű szimbólumok neve valójában @ nélkül kerül fordításra. Így attól függetlenül, hogy egy másik nyelv nem tartalmazza például a sealed kulcsszót, és egy programban azonosítóként használják, C#-ban lehetőség van az azonosító közvetlen használatára.

2005.02.24.

12

Mi a megengedett szintaxis?

- A leggyakrabban:
letter{letter|digit}
néha az '_' beszúrása is megengedett.
- **Pascal**: letter{letter|digit}
- **ADA95**: letter{[_]letter|[_]digit},
 - itt a betűk halmaza nagyobb, így pl. Magasság OK., de problémás az 'ö' és az 'ü'.
- **Java**: Java_letter{[_]Java_letter|[_]digit},
 - αβγδε vagy El_Niño is OK.

2005.02.24.

13

Van-e hosszúsági megkötés az azonosítókra?

- **Fortran66, Fortran77**: max. 6 jel
- **Fortran90**: max. 31 jel
- **ADA**: be kell férjen egy sorba...
- **Pascal, Java**: tetszőleges hosszúság

2005.02.24.

14

Vannak-e fenntartott szavak?

- **Pascal, C, C++, Java, Perl, CLU, ADA95** etc.:
 - a fenntartott szavak nem használhatók azonosítóként
- **ADA95**:
 - különbséget tesz a kulcsszavak és az elődefiniált szavak között, mint pl. : Integer, True, etc..
 - Átdefiniálhatók:
type Integer is range -999_999..+999_999;
így a program implementáció-független
de: True: Integer ... -- nincs értelme...
- **Fortran, PL/1**:
megengedett a kulcsszavakat azonosítóként használni!

2005.02.24.

15

Literálok

- Milyen numerikus literálok vannak?
- Milyen más alapok megengedettek a 10-esen kívül?
- Mi az egész, ill. valós literálok szintaxisa?
- Többsoros sztring literálok megengedettek-e?
- Vezérlőkarakterek sztring literálokban megengedettek-e, és hogyan írjuk le őket?

2005.02.24.

16

Numerikus literálok:

- A gyakorlatban csak a '0'..'9' és az 'A'..'F'
- néha '_'
- '' (space) soha :-(

Mi a számok szerkezete?

- **Egészek**: decimális: [-]digit{digit} -123, 456789
pl. **ADA95, Perl, Eiffel**: '_' is: 456_789
ADA95-ben: exponenciális alak is - pl. 1E6 -, ahol a kitevő nem negatív
C, C++, Java, C#: *int* és *long* különbsége: 'L' a literál végén, pl. 2 vagy 2L. 'U' az unsigned

2005.02.24.

17

• Valósak:

[-]digit{digit}.digit{digit}[exponent]
-123.456E+3

Majdnem minden nyelv ad ehhez valami specialitást:

Pascal: [-]digit{digit}[exponent] 12E+3

ADA95: '_' is lehet: -1_234.0

Eiffel: [-]{digit}.{digit}[exponent] -1.

Java: "float_suffix" is lehet a végén

2005.02.24.

18

Milyen alapok megengedettek?

- Általában: decimális számok
- néha más is -- pl. 2, 8 16
- **Pascal, Modula, Oberon**: csak decimális számok
- **C, C++, Java**: Oktális, decimális és hexadecimális
oktális: 0734
hexadecimális: 0xFF, 0xC0B0L.
- **ADA95**: minden alap 2 és 16 között:
base#mantissa#[exponent]
pl. 16#FFF.F#E1
- **Mathematica**: minden alap 2 és 36 között!
(minden számjegy és angol betű OK.) :
basis ^^mantissa
pl.:30^^Mathematica = 13 207 019 439 499 570₁₀

2005.02.24.

19

Karakterek és karaktersorozatok

- Karakterek: 'A' vagy „escape szekvenciák”:
'\n', '\", 'ooo' stb.
- Sztringek "A" - tárolásuk különböző lehet.

2005.02.24.

20

Megjegyzések

- Szerkezete befolyásolja a program megbízhatóságát
- Szokásos lehetőségek:
 - egy speciális oszloptól (jeltől) a sor végéig
 - pl. Fortran 'C' az első oszlopba
 - speciális jel(ek) elején - végén
 - speciális jel(ek) elején - vége a sor vége

2005.02.24.

21

Megjegyzések

- **Pascal**: (* és *) vagy { és }
- **CLU**: %-tól sor végéig
- **C**: /* és */
- **C++**: // -tól sor végéig
vagy /* és */
- **Java**: mint C++-ban, vagy:
/** documentation */
- **C#**: mint C++-ban, vagy:
/// egysoros dokumentációs megjegyzés
- **ADA95, Eiffel**: -- -tól sor végéig

2005.02.24.

22