

Imperatív programozás

Programozási nyelvek szabályrendszere



Kozsik Tamás és mások

ELTE Eötvös Loránd Tudományegyetem

- 1 Programozási nyelvek definíciója
- 2 Kitekintés későbbi tárgyra
- 3 Pragmatika

Programozási nyelv szabályai

- Lexikális
- Szintaktikus
- Szemantikus



Lexikális szabályok

Milyen építőkövei vannak a nyelvnek?

- Kulcsszavak: while, for, if, else stb.
- Operátorok: +, *, ++, ?: stb.
- Zárójelek és elválasztó jelek
- Literálok: 42, 123.4, 44.44e4, "Hello World!" stb.
- Azonosítók
- Megjegyzések

Case-(in)sensitive?



Azonosító

- Alfánumerikus
- Ne kezdődjön számmal
- Lehet benne _ jel?

Jó

- `factorial`, `i`
- `computePi`, `open_file`, `worth2see`, `Z00`
- `__main__`

Rossz C-ben

- `2cents`
- `fifty%`
- nőnemű és *Αθήνα* (jók Pythonban)



Szintaktikus szabályok

Hogyan építkezhetünk?

- Hogyan épül fel egy ciklus vagy egy elágazás?
- Hogyan néz ki egy alprogram? stb.



Backus-Naur form (Backus normal form) – BNF

```
<statement> ::= <expression-statement>  
              | <while-statement>  
              | <if-statement>  
              | ...
```

```
<while-statement> ::= while (<expression>) <statement>
```

```
<if-statement> ::= if (<expression>) <statement>  
                  <optional-else-part>
```

```
<optional-else-part> ::= "  
                        | else <statement>
```



Szemantikus szabályok

Értelmes, amit építettünk?

- Deklaráltam a használt változókat? (C)
- Jó típusú paraméterrel hívtam a műveletet?

stb.



Típusellenőrzés

- A változókat, függvényeket a típusuknak megfelelően használtuk-e
- A nem típushelyes programok értelmetlenek

Statikus és dinamikus típusrendszer

- A C fordító ellenőrzi *fordítási időben* a típushelyességet
- Pythonban *futási időben* az interpreter vizsgálja ezt

Erősen és gyengén típusos nyelv

- Gyengén típusos nyelvben automatikusan konvertálódnak értékek más típusúra, ha kell
 - Eleinte kényelmes
 - De könnyen írunk mást, mint amit szerettünk volna
- A C-ben és Pythonban viszonylag szigorúak a szabályok (erősen típusosak)

Statikus és dinamikus szemantikai szabályok

- Statikus: amit a fordító ellenőriz
- Dinamikus: amit futás közben lehet ellenőrizni

Eldönthetőségi probléma...



Összefoglalva

- Lexikális: mik az építőkövek?
- Szintaktikus: hogyan építünk struktúrákat?
- Szemantikus: értelmes az, amit felépítettünk?
 - statikus szemantikai szabályok
 - dinamikus szemantikai szabályok



- 1 Programozási nyelvek definíciója
- 2 Kitekintés későbbi tárgyra
- 3 Pragmatika

A fordítóprogramok részei

- Lexer: tokenek sorozata
- Parser: szintaxisfa, szimbólumtábla
- Szemantikus (pl. típus-) ellenőrzés

(vagy különböző szintű fordítási hibák)



Formális nyelvek

- Lexikális szabályok: reguláris nyelvtan
- Szintaktikus szabályok: környezetfüggetlen nyelvtan
- Szemantikus szabályok: környezetfüggő, vagy megkötés nélküli nyelvtan



Program szemantikája

A (nyelv szabályainak megfelelő) program jelentése



- 1 Programozási nyelvek definíciója
- 2 Kitekintés későbbi tárgyra
- 3 Pragmatika

A nyelv definíciója

- Lexika
- Szintaktika
- Szemantika
- Pragmatika



Pragmatika

Hogyan tudjuk hatékonyan kifejezni magunkat?

- Konvenciók
- Idiómák
- Jó és rossz gyakorlatok

stb.



Konvenció

általános vagy fejlesztői csoportra (cégre) specifikus

- kapcsos zárójelek elhelyezése
- névválasztás (pl. setter/getter)
- azonosítók írásmódja, nyelve
- kis- és nagybetűk

