



Programozási nyelvek a  
közoktatásban  
5. előadás



## Neumann-elvű nyelvek

### 1. Elképzelések, tulajdonságok

- memória, címezhető (sorszámmal)
- a program és az adatok a memóriában vannak
- a végrehajtás memóriaállapotok sorozata, állapottér (adatok tere), koordináták, programfüggvény: állapottér  $\rightarrow$  állapottér, konkrét állapot=pont,
- a program leírása szöveges







## Neumann-elvű nyelvek

### 2. Következmények

- van változó (névvel elnevezett memória tartomány)
- van értékadás
- utasítások ismételt végrehajtása lehetséges (ciklus, GOTO)
- elágazás (GOTO)
- eljárás (adott címen levő program végrehajtása, visszatérés)
- beolvasás: memória másolás (pl. billentyűzet puffer)
- kiírás: memória másolás (pl. képernyő)





## Automata-elvű nyelvek

### 1. Elképzelések, tulajdonságok

- az adatok állapotok, illetve bemenetek – fix felosztású memória (pl. festőkar helye, fest-e, ...): állapottér
- állapotkomponensek
- a végrehajtás egyszerű állapotok sorozata
  - a program egy állapotátmenet-függvény
  - a program elkülönül az állapotoktól
- tevékenységorientáltság (állapotváltoztatás, állapotlekérdezés)
- a végrehajtó egy (véges) automata



Pé. ipari robotok, Logo teknőc



## Automata-elvű nyelvek

### 2. Következmények

- az utasítások általában egy állapotkomponenst változtatnak
- változó, értékadás nincs (csak névvel ellátott állapotkomponensek)
- csak primitív (állapot-, vagy paraméter- vagy érzékelő-állapotfüggő) ciklusok írhatók
- állapot-, vagy paraméter vagy érzékelő-állapotfüggő elágazás





## Automata-elvű nyelvek

### 2. Következmények

- utasítások paraméterezhetőek
- eljárások lehetnek, paraméterezhetőek
- beolvasás: paraméterezés, állapotlekérdezés, érzékelő állapota
- kiírás: állapotváltozás nyoma
- párhuzamosság – több végrehajtó automata





# Számítási modellek



## Funkcionális (függvényszerű) nyelvek

### 1. Elképzelések, tulajdonságok

- a program egy függvény
- memória nincs
- a végrehajtás függvénykifejtés
- erős matematikai kidolgozottság





## Funkcionális (függvényszerű) nyelvek

### 2. Következmények

- változó, értékadás nincs
- konstansok = konstans függvények
- függvénykompozíció
- feltételes függvények
- ciklus nincs, helyette rekurzív függvények (esetleg lehet ciklus, ha léteznek a programon értelmezett függvények)







# Számítási modellek



## Funkcionális (függvényszerű) nyelvek

Függvénykompozíció:  $f(x) := g \circ h(x)$

Alternatív függvény:  $f(x) := \begin{cases} g(x) & \text{ha } p(x) \\ h(x) & \text{ha } \neg p(x) \end{cases}$

Rekurzív függvény:  $f(x) := \begin{cases} g(x) & \text{ha } p(x) \\ h \circ f \circ i(x) & \text{ha } \neg p(x) \end{cases}$





## Funkcionális (függvényszerű) nyelvek

### 2. Következmények

- függvényargumentum van
- különböző stratégiák a paraméterátadásra
- beolvasás: speciális függvény
- kiírás: a függvény eredménye, esetleg speciális (mellékhatással rendelkező függvény)





## Logikai nyelvek

### 1. Elképzelések, tulajdonságok

- a program egy logikai formula (általában nem lehet teljesen általános formula: megszorítások!)
- a program végrehajtása a logikai formula kiértékelése
- erős matematikai kidolgozottság
- a formulákban szerepelhetnek paraméterek
- a program és az adatok nem különülnek el (minden adat egy azonosan igaz formula)





# Számítási modellek



## Logikai nyelvek

### 2. Következmények

- nincs változó, nincs értékadás
- szekvencia = és, elágazás = vagy (pontosabban ezek megfelelő alkalmazása)
- ciklus nincs, helyette rekurzív formulák
- vannak feltételes kifejezések
- elemi utasítás: mintaillesztés







# Számítási modellek



## Logikai nyelvek

### 2. Következmények

- beolvasás, kiírás: mintaillesztés
- lehetőség új formulák létrehozására (új utasítás), formulák törlésére
- új formulák adatbázisba mentése
- kiértékelési algoritmus: backtrack (a felesleges visszalépések levágásával)





Programozási nyelvek a  
közoktatásban  
5. előadás vége