
3.3. Algoritmusok fogalma, algoritmuskészítés módszerei leírónyelvek Folyamatábrázolási módszerek

Definíciók:

Algoritmus: bármely véges sok lépéssel leírható tevékenység. Olyan pontos előírás, amely megmondja, hogy egy adott típushoz tartozó feladat megoldásához milyen műveleteket kell meghatározott sorrendben elvégezni

Lépés: Közvetlenül elvégezhető (program utasítássá átírható) vagy algoritmusként kezelhető elem

Az algoritmussal szemben támasztott követelmények:

- Leírónyelv (szabványosított)
- Egyértelmű eredményt szolgáltatasson
- Jól meghatározott értelmezési tartomány
- Helyes
- Véges lépésszámon belül *érjen véget*
- *Általános:* ne csak egy konkrét feladatot oldjon meg, hanem lehetőleg minél szélesebb problémaosztályt

Az algoritmusok leírásának eszközei:

1. **Folyamatábra**
2. **Struktogram**
3. **Mondatszerű leírás**

Cél:

A feladatok megoldásainak leírása programozási nyelvtől független nyelven.

Példafeladatunk:

N tanuló év végi tanulmányi átlagának ismeretében adjuk meg a jeles átlagú tanulók számát.

1. Folyamatábra

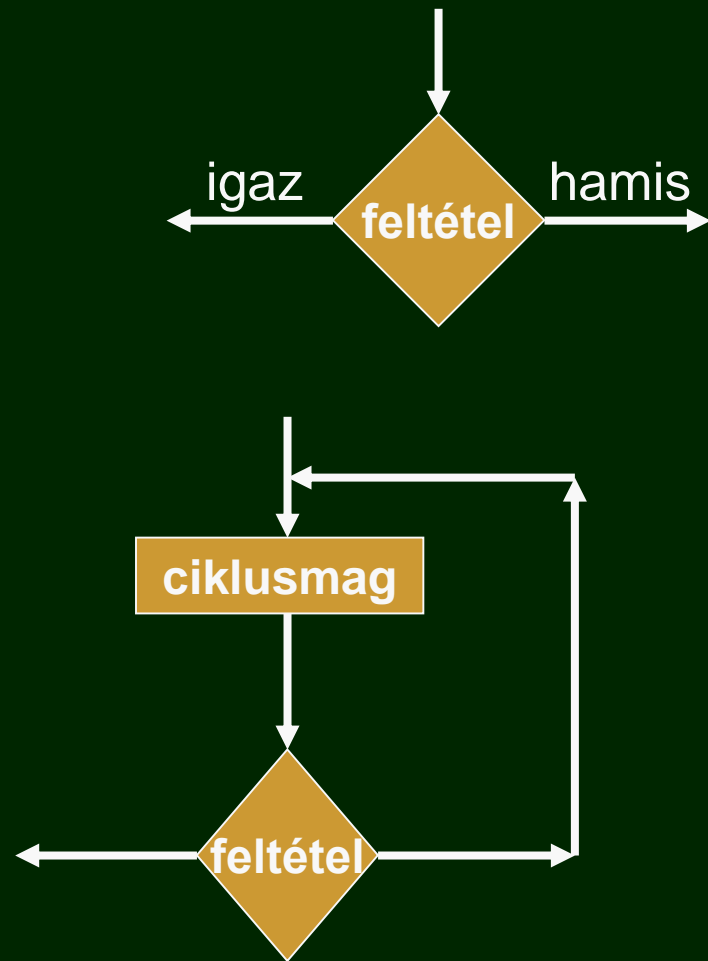
- Legkorábbi
 - A programot gráfként írja le
- A programgráf egy irányított gráf: **csomópontokból**, és őket összekötő **élekből** áll.
 - Egyetlen kiinduló és egyetlen befejező éle van.
 - Az induló élből bármely csomópont elérhető, s bármely csomópontból el lehet jutni a befejező élre.

Háromféle csomópont:

1. **Utasítás** csomópont (függvény csp)
2. **Döntés** csp
3. **Gyűjtő** csp

Program: A fenti három csomópontból felépített gráf.

Végrehajtás: A programgráf csomópontjainak bejárása irányított élek mentén



Algoritmus struktúrák (elemei)

- **Szekvencia:** egymás után végrehajtandó tevékenységek sorozata
- **Elágazás (szelekció):** feltételes elágazások, választás megadott tevékenységek közül
- **Ciklus (iterációk):** megadott tevékenységek ismételt végrehajtása. A ciklusmagot nullszor, egyszer vagy sokszor hajtjuk végre a ciklus feltételétől függően.

Definíció:

Strukturáltnak nevezzük azt a **programot**, amely csak a fenti három algoritmus szerkezetet tartalmazza.

Előnye:

Az algoritmus végrehajtása könnyen követhető rajta

Hátránya:

Terjedelmes

Szövegszerkesztővel nehezen készíthető

Javítása nehézkes

Nagy program esetén áttekinthetetlen

Nem szabályos struktúrák is létrehozhatóak vele

Egyéb csomópontok:

4. Start csomópont
5. Stop csomópont
6. Input-output csomópont
7. Ciklus csomópont

Példafeladat

2. Struktogram

A programgráfot élek nélkül ábrázoljuk, így alapeleme:

Utasítás

Alapszerkezetei:

- Szekvencia
- Szelekció
- Iteráció (elől tesztelő!)

Minden utasítás doboz helyén állhat:

- Egyetlen utasítás
- A három algoritmus szerkezet valamelyike
- Eljáráshívás

Egyéb szerkezetek:

Eljárás definíció

Sokirányú elágazás

Hátul tesztelő ciklus

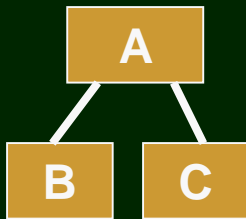


Jackson diagramm

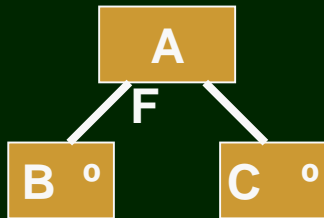
Adat és algoritmus szerkezet leírására egységes ábrakészletet definiál.

Jackson féle jelölés:

-szekvencia:



-szelekció:



Funkcionális leírás (pseudokód):

A seq

B

C

A end

A select F

B

A or (not F)

C

A end

-iteráció:



A iter while F

B

A end

Előnye:

Algoritmus és adatlírás egységessége

Hátránya:

Még áttekinthetlenebb

Leírás absztrakt függvényekkel

Matematikai eszköz, formulákkal írja le a programot.

Elemei:

- Szekvencia: $P = \text{SEQ}(Q, R)$
- Elágazás: $P = \text{IF}(f, Q, R)$
- Ciklus: $P = \text{DO}(f, Q)$ vagy $P = \text{UNTIL}(Q, f)$
- f : logikai formula
- Q, R : utasítás

-többágú szelekció:

Elágazás

feltétel 1 esetén utasítás 1

feltétel 2 esetén utasítás 2

:

feltétel n esetén

egyébként utasítás n+1

Elágazás vége

Iteráció: -Számláló:

Ciklus *cv*= *kezdet*-től *vég*-ig *n*-esével
ciklusmag

Ciklus vége

-Elöl tesztelő:

Ciklus *amíg feltétel*
ciklusmag

Ciklus vége

- Hátultesztelő: **Ciklus ciklusmag**
Amíg feltétel
Ciklus vége

Program, eljárás illetve függvény leírása:

Program:

utasítások

Program vége.

Eljárás Eljárásnév (formális paraméterlista):

utasítások

Eljárás vége.

Függvény Függvénynév (formális paraméterlista): típus

utasítások

függvénynév:=kifejezés

Függvény vége.

Adatleírás:

Változók: *Változó* név : típus

pl.: Változó szám : egész

Konstansok: *Konstans* név : típus (kezdőérték)

pl.: Konstans szám : egész(7)

Típusok: *Típus* típusnév = (struktúra)

pl.: Típus TömbTíp = Tömb(1..10 : egész)

Értékadások:

Változó név := kifejezés

Kifejezés kiértékelése:

Precedencia sorrend:

1. Függvények
2. 1 operandusú kifejezések
3. 2 operandusú kifejezések
4. Relációk

Mindvégig a balról-jobbra haladást követjük, melyet a zárójelezés módosíthat.

pl. $3 + 5 * 4 - \text{sqrt}(3) > 7 + (2 - 4^2)$

A műveletek lehetnek:

- Unáris (egy operandusú)
pl.: -7
- Bináris (két operandusú)
pl.: 4+2

Az operátorok elhelyezkedése alapján:

- **Prefix:** A műveleti jel az operandus *előtt* van
- **Postfix:** A műveleti jel az operandus *után* van
- **Infix:** A műveleti jel az operandusok *között* helyezkedik el.

Feladatok

- Döntsd el egy számról, hogy páros-e!
- Döntsd el, három szakaszból, hogy szerkeszthető-e belőle háromszög!
- 100 beolvasott szám között hány páros van?
- Olvass be számokat addig, míg összegük el nem éri 100-at.
- 10 véletlen szám összege, vagy szorzata nagyobb?
- Döntsd el egy számról, hogy prím-e!
- Add meg egy szám összes osztóját!