

Vezérlési szerkezetek és főbb jellemzőik

- 1) **Szekvencia:** egymás után írt utasítások sorozata. A program egymást követően hajtja végre őket
- 2) **Szelekció – elágazás:** adott feltétel teljesülésétől függően írja elő utasítások végrehajtását.
Fajtái:
 - **IF – EGYÁGÚ:** ha adott feltétel teljesül, akkor végrehajt egy/több utasítást (utasításblokkot)
 - **IF-ELSE – kétágú:** meghatározza, hogy ha egy adott feltétel teljesül, milyen utasítás(ok)t hajtson végre, illetve azt is, hogy nem teljesülés esetén mi kerüljön végrehajtásra
 - **többágú:** több feltétel szerint lehet végrehajtani
 - ✓ **IF – ELSE IF – ELSE:** értékhatárok közötti vizsgálat
 - ✓ **SWITCH – CASE:** konkrét értékre mit hajtson végre
- 3) **Ciklusok:** adott utasítás(ok) feltételtől függő ismételt végrehajtását eredményező vezérlés. Az utasításokat egyszer specifikáljuk, de többszöri végrehajtásuk lehetséges.

Ciklus részei:

- vezérlő feltétel (ciklusfej – előtesztelőnél, ciklusvég – hátulatesztelőnél): a feltétel teljesülése esetén a ciklusban marad a program
- ciklusmag: amely utasításokat szeretnénk ismételni

Fajtái:

- előtesztelő
- hátulatesztelő
- léptető

WHILE ciklus – előtesztelő ciklus

Előtesztelő ciklus, azaz először a ciklus feltételét vizsgálja meg, hogy teljesül-e, ha igen, akkor végrehajtja a ciklusmagban megadott utasításokat. Mindaddig a ciklusban marad, amíg a feltétel teljesül. A ciklusmag lehet, hogy egyszer sem kerül végrehajtásra. Csak óvatosan, mert könnyű belefutni egy végtelen ciklusba!

DO...WHILE ciklus – hátulatesztelő ciklus:

Először a ciklusmag utasításait hajtja végre, ezután megvizsgálja a ciklus feltételét, és ha teljesül, akkor a ciklusban marad. A hátulatesztelő ciklus magja egyszer biztosan végrehajtásra kerül.

A léptető ciklus – FOR ciklus

A léptető ciklus esetén tudjuk, hogy hányszor kell végrehajtani a ciklusmag utasításait. Ezt nevezük ciklusváltozónak, melynek megadjuk a kezdőértékét, a végértékét és a lépésközét.

Összegzés tétele: egy sorozathoz egy értéket rendelő algoritmus

- Feladat: egy sorozat elemeinek összege
- Bemenet: $X[N]$ – N elemű tömb
- Kimenet: S – összeg
- Algoritmus:

```
Összegzés( $X[N]$ ,  $S$ )  
     $S=0$ ;  
    Ciklus  $i=1$ -től  $N$ -ig  
         $S=S+X[i]$ ;  
    Ciklus vége;  
Eljárás vége;
```

- Alkalmazása: összegzés (szumma), átlagszámítás

Maximum kiválasztás: egy sorozathoz egy értéket rendelő algoritmus

- Feladat: egy sorozatban megtalálni legnagyobb elem helyét
- Bemenet: $X[N]$ – N elemű tömb
- Kimenet: \max – egész
- Algoritmus:

```
Maximum_kiválasztás( $X[N]$ ,  $\max$ )  
     $\max=1$ ;  
    Ciklus  $i=2$ -től  $N$ -ig  
        Ha  $X[i]>X[\max]$ , akkor  
             $\max=i$ ;  
        Elágazás vége;  
    Ciklus vége;  
Eljárás vége;
```

Eldöntés tétele: sorozathoz logikai értéket rendelő algoritmus

- Feladat: Egy sorozatban van-e legalább egy darab T tulajdonságú elem
- Bemenet: $X[N]$ – N elemű tömb
 T tulajdonság
- Kimenet: VAN – (logikai)

- Algoritmus

```

Eldöntés(X[N],T, VAN)
    i:=1;
    Ciklus amíg i<=N és X[i] nem T tulajdonságú
        i=i+1;
    Ciklus vége;
    Ha i<=N akkor
        VAN
    Elágazás vége;
Eljárás vége;

```

Megszámlálás tétele: egy sorozathoz egy egész értéket rendelő algoritmus

- Feladat: Egy sorozatban hány darab T tulajdonságú elem van
- Bemenet: X[N] – N elemű tömb
T tulajdonság
- Kimenet: DB – darabszám (egész!)
- Algoritmus:

```

Megszámlálás(X[N],T)
    DB=0;
    Ciklus i=1-től N-ig
        Ha X[i] T tulajdonságú, akkor
            DB=DB+1;
    Elágazás vége;
    Ciklus vége;
Eljárás vége;

```

Minimum kiválasztás: egy sorozathoz egy értéket rendelő algoritmus

- Feladat: egy sorozatban megtalálni a legkisebb elem helyét
- Bemenet: X[N] – N elemű tömb
- Kimenet: min – egész

- Algoritmus:

```

Minimum_kiválasztás(X[N], min)
    min=1;
    Ciklus i=2-től N-ig
        Ha  $X[i] < X[\text{min}]$ , akkor
            min=i;
        Elágazás vége;
    Ciklus vége;
Eljárás vége;

```

Keresés tétele: egy sorozathoz egy értéket rendelő algoritmus

- Feladat: Egy sorozatban hányadik helyen található T tulajdonságú elem, melyről nem tudjuk, hogy szerepel-e a sorozatban
- Bemenet: $X[N]$ – N elemű tömb
T tulajdonság
- Kimenet: sorszám
- Algoritmus:

```

Keresés(X[N], T, sorszám)
    i:=1;
    Ciklus amíg  $i \leq N$  és  $X[i]$  nem T tulajdonságú
        i:=i+1;
    Ciklus vége;
    Ha  $i \leq N$  akkor
        sorszám:=i;
    Különben
        sorszám:=-1;
    Elágazás vége;
Eljárás vége;

```

Kiválasztás tétele: egy sorozathoz egy értéket rendelő algoritmus

- Feladat: Egy sorozatban hányadik helyen található T tulajdonságú elem, melyről tudjuk, hogy szerepel a sorozatban
- Bemenet: $X[N]$ – N elemű tömb
T tulajdonság
- Kimenet: sorszám
- Algoritmus:

```
Kiválasztás( $X[N]$ , T, sorszám)
    i:=1;
    Ciklus amíg  $X[i]$  nem T tulajdonságú
        i:=i+1;
    Ciklus vége;
    sorszám:=i;
Eljárás vége;
```

Kiválogatás tétele: egy sorozathoz egy vagy több sorozatot rendelő algoritmus

- Feladat: kiválogatni a T tulajdonságú elemeket
- Bemenet: $X[N]$ – N elemű tömb
T tulajdonság
- Kimenet: $Y[M]$ – M elemű tömb
- Algoritmus:

```
Kiválogatás( $X[N]$ , T,  $Y[M]$ )
    számláló:=1;
    Ciklus i:=1-től N-ig
        Ha  $X[i]$  T tulajdonságú akkor
             $Y[\text{számláló}]=X[i]$ ;
            számláló:=számláló+1;
    Elágazás vége;
    Ciklus vége;
Eljárás vége;
```