**2. modul**

Programming Basics

Programming Basics

oop basics, programming basics, java basics, java core, junit, sql basics

# Algoritmus

Lásd [Algoritmusok és adatszerkezetek](https://drive.google.com/file/d/1kQFvQl2eoVdLf-vmGoj5PCpuXtoeaVhm/view?usp=share_link) SZTE mérnök info tananyag részlet

# Vezérlési szerkezetek

A magas szintű programozási nyelvek használata a modern programozás alapja. Az emberek sokkal könnyebben átlátnak olyan típusú leírásokat, amelyek közelebb állnak az emberi gondolkodáshoz, kifejezési formákhoz, mint mondjuk egy gépi kódban leírt számsorozatot.

Az emberi korlátok közül egy fontos tényező korlátozott memóriánk. Egyszerű tesztekkel igazolható, hogy az emberek átlagosan egyszerre körülbelül 7 dolgot tudnak észben tartani.

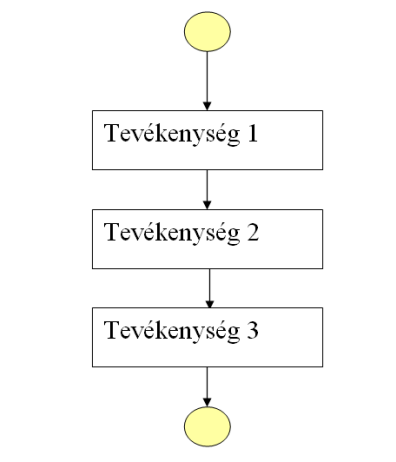
Ha egy olyan algoritmust készítünk, amely igen sok alkotórészből áll, akkor ez előbb vagy utóbb áttekinthetetlenné válik, mind a készítője, de különösen más olvasók számára. Célszerű tehát törekedni arra, hogy algoritmusaink viszonylag kevés építőelemből álljanak.

## Tevékenységsorozatok

Tevékenységsorozat, más néven **szekvencia**. Egyszerűen azt jelenti, hogy különféle tevékenységeket végzünk, mégpedig egyiket a másik után.

**Például:**

* Forralok vizet, azután beleteszem a teafiltert, majd két perc múlva kiveszem, végül öntök bele egy kis citromlevet.



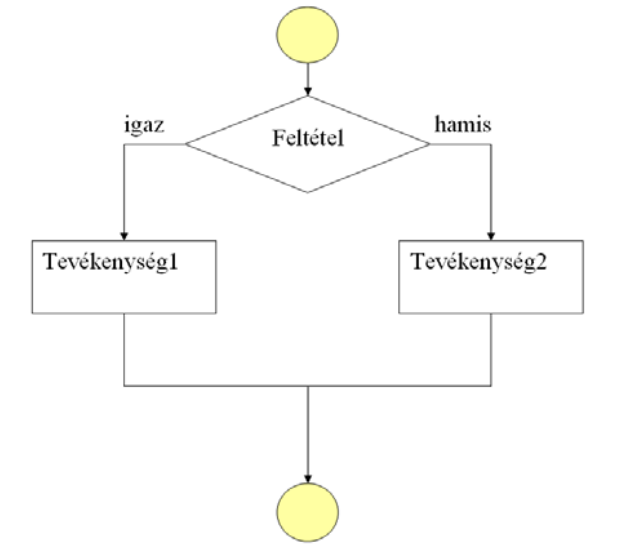
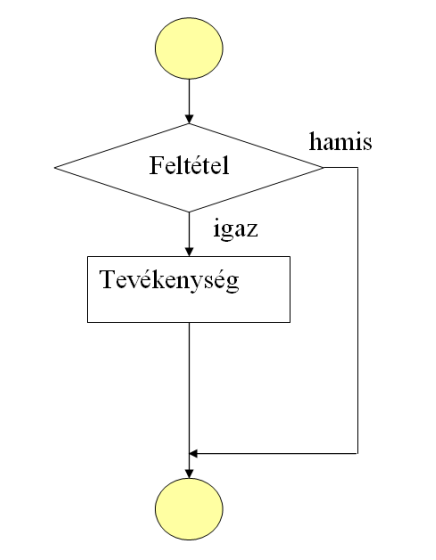
## Elágazások

Elágazás, más néven **szelekció**. Gyakran tevékenységeink valamilyen feltétel kiértékelésének eredményétől függenek. A hétköznapi életben a választások, döntések mind szelekcióval írhatók le. Tehát segítségével alternatív végrehajtási módokat tudunk leírni.

A szelekció egyik fontos eleme a **feltétel**, amelyet kiértékelünk a szelekció végrehajtása előtt. Ennek eredményétől függ, hogy a lehetséges tevékenységek közül **melyik** tevékenységet hajtsuk végre.

**Például**:

* **HA** sikerül időben lefeküdni aludni, **akkor** reggel elmegyek futni. **Egyébként** pedig este sétálok egy kicsit.



## Ciklusok

Ciklus, más néven iteráció. Nem szolgál másra, mint az ismétlődő tevékenységek leírására. A mindennapi életben is gyakran alkalmazunk ismételt tevékenységeket.

**Például**:

* Addig jár a korsó a kútra, amíg el nem törik

A ciklusok mindig tartalmaznak egy tevékenységet, valamint egy feltételt, aminek függvényében tovább folytatjuk vagy abbahagyjuk a tevékenység ismétlését. Attól függően, hogy mikor végezzük el a feltétel kiértékelését, kétféle ciklusokról beszélünk:

* Elöltesztelő ciklus
* Hátultesztelő ciklus

**Például:**

* **(Előltesztelő)** Beleharapjak ebbe a fánkba? Amennyiben a válasz igen, akkor harapok egyet. Ismét felteszem a kérdés: Beleharapjak ebbe a fánkba? Ezt mindaddig ismétlem, amíg el nem fogy a fánk.
* **(Hátultesztelő)** Először beleharapok a fánkba, majd felteszem a kérdést: Jó ez a fánk még? Amennyiben a válasz igen, akkor megint beleharapok, de ismételtem felteszem a kérdést. Addig ismételve, amíg a válasz egyszer csak nem lesz.

### 

### 

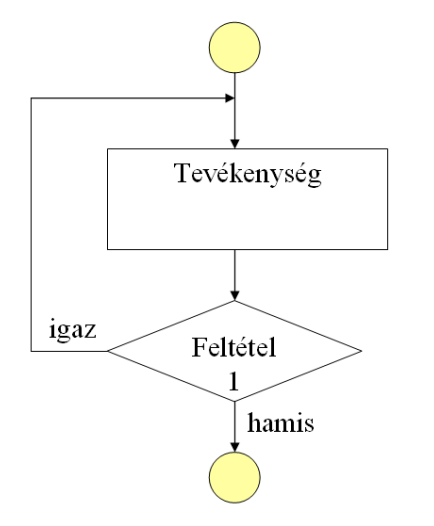
### 

### 

### Elöltesztelő

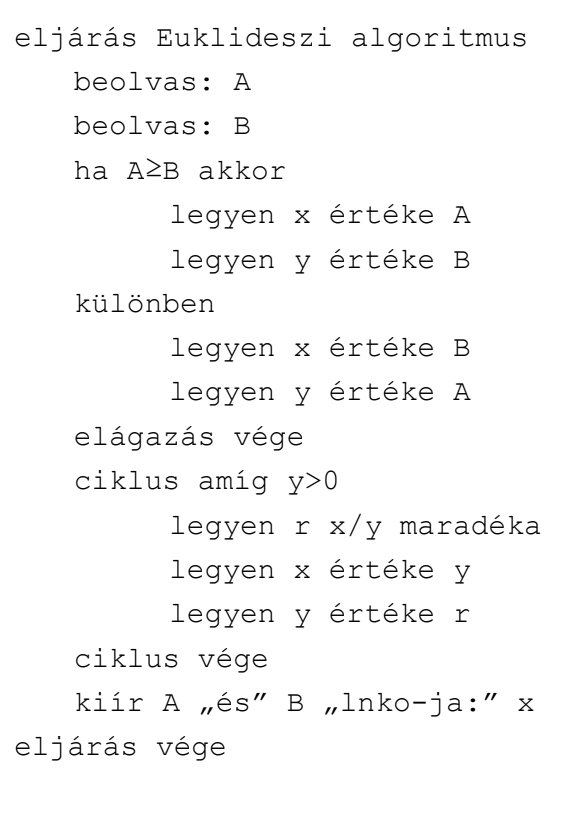
### 

### Hátultesztelő



Amennyiben van olyan tevékenység, amit szeretnénk, hogy minimum egyszer végrehajtásra kerüljön akkor használjunk hátultesztelő ciklust, egyébként használjunk elöltesztelő ciklus.

# Kódstruktúra



# Feladatok

1. (Órai) Írjunk algoritmust a feltöltőkártyás telefon egyenleg feltöltésére
2. Készíts algoritmust a limonádé készítésére? (minimum 7 lépés)
3. (Órai) Írjunk algoritmust rántotta készítésére, mely tartalmaz egy elágazást és egy ismétlést is
4. (Órai) Írjunk algoritmust XY-ra, mely tartalmaz egy elágazást és egy ismétlést is
5. Írj egy példát egy legalább 7 lépést, egy elágazást és egy ismétlést tartalmazó algoritmust önállóan.
6. Készítsetek algoritmust az ATM-ből való készpénzfelvételre, mint szegény ember. Az algoritmus tartalmazza a következő készpénzfelvételi lehetőségeket. 5.000, 10.000, 25.000, 50.000, 100.000 és egyéb.

# Linkek

Simon Gyula SZTE - [Programozás Alapjai](https://drive.google.com/file/d/1H55zAi7w4N0ACqjMypB36EjRXAWUoS8-/view?usp=share_link)