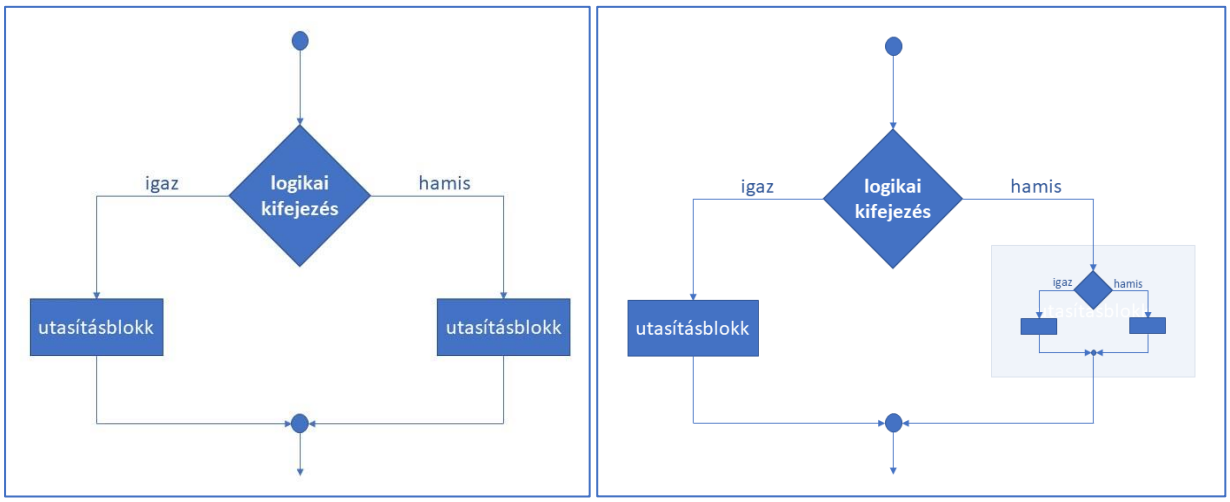
Vezérlőszerkezetek

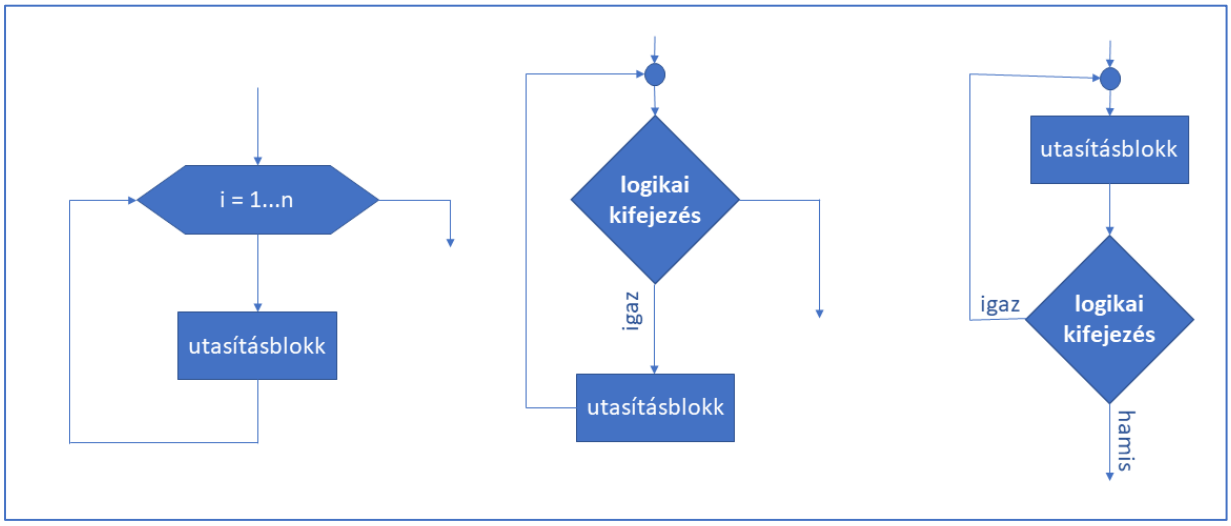
A vezérlő- vagy programszerkezetek olyan algoritmuselemeket jelentenek, melyek a legtöbb ún. magasszintű programozási nyelvben azonos elven működnek, csak más szintaktikával írjuk le őket. Itt és most általánosságban nézzük meg ezeket egy algoritmusleíró eszköz segítségével, azaz folyamatábrákkal:

1. Logikai elágazás (Szelekció)



Az elágazások lényege, hogy a program utasítássorozata egy logikai kifejezés igaz vagy hamis értéke alapján egyik vagy másik irányban folytatódhat. Ez megfelel a hétköznapi (vagy matematikai logikai) „ha … akkor… különben” szerkezetű mondatnak. A „ha” után van egy állítás (logikai kifejezés), az „akkor” az igaz-ág, és a „különben” a hamis-ág. Ez utóbbi sokszor el is maradhat. A fenti jobboldali ábra azt illusztrálja, hogy ezek a szerkezetek egymásba ágyazhatók.

2. Ismétlődés (Iteráció, ciklus)



számlálós ciklus, előfeltételes ciklus, utófeltételes ciklus

Az ismétlődések célja, hogy sok elemmel ugyanazt a műveletsort elvégeztessük. Nyilván, ha a műveletsort például tízszer leírjuk, az is jó, de nem praktikus. Erre találták ki ezeket a vezérlőszerkezeteket.

A számlálós ciklusok adott számszor hajtják végre az utasításokat, a feltételes ciklusok pedig egy logikai kifejezés igaz értéke esetén (vannak nyelvek, ahol van olyan ismétlődésfajta, ami pont fordítva „gondolkodik”). Az utófeltételes lényeges eltérése az előfeltételestől, hogy egyszer mindenképpen végrehajtja a ciklusmagot (az utasításblokkot)!

Nem csak ezek a szerkezetek vannak, de számunkra most ezek a fontosak.

Szoftverek letöltése, szoftverlista

Python

Elágazások (szelekciók)

```
if_else_Python.py - C:/Users/Opus/Documents/if_else_Python.py (3.10.4)
File Edit Format Run Options Window Help
eleje='A megadott szám'
szam=int(input('Kérek egy egész számot: '))

if szam%2==0:
    print(eleje,'páros')
else:
    print(eleje,'páratlan')
```

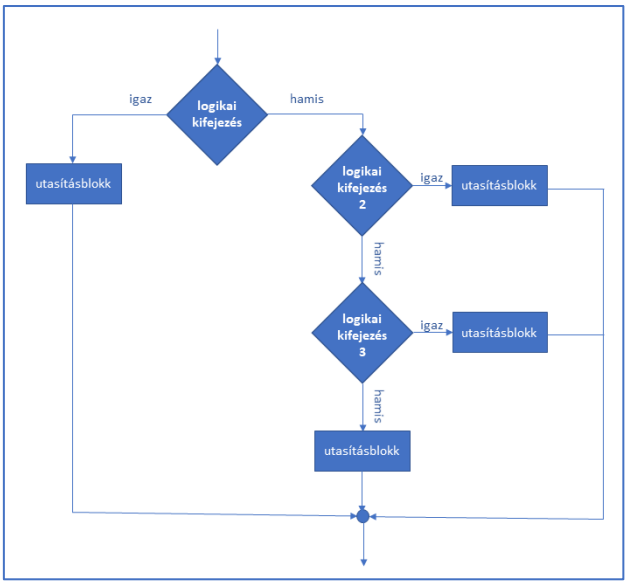
C++

Elágazások (szelekciók)

```
Start here | vezerloszerkezetek_P_vs_C.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      setlocale(LC_ALL,"hun");
8
9      int szam;
10     string eleje="A megadott szám ";
11     cout << "Kérek egy egész számot: ";
12     cin >> szam;
13
14     if(szam%2==0)
15     {
16         cout << eleje+"páros" << endl;
17     }
18     else
19     {
20         cout << eleje+"páratlan" << endl;
21     }
22
23     return 0;
24 }
25
```

További lehetőségek

A fenti képeken egy alaphelyzet látható. Sőt, az ún. „hamis-ág” sokszor el is marad, ha nincs rá szükség. Van viszont, mindkét nyelvben arra lehetőség, hogy a hamis-ágon újabb igaz-hamis ágra bontsuk a program menetét (igaz-ágon is lehet további elágazás, természetesen). Lássuk:



Python	C++:
if… elif… else	if… else if… else

Szoftverek letöltése, szoftverlista

Érték szerinti elágazás

Eddig a logikai elágazásokról volt szó, de a C/C++-ban van egy speciális vezérlőszerkezet, amely a Python 3.10 és későbbi verziókkal a Pythonban is használható, amit a következővel illusztrálhatok:

```
Van egy hely, akármilyen, ahol vannak kapcsolók, akármi-
lyenek, melyek mondjuk 1-től 5-ig vannak számozva, és a
következő dialógusnak lehetünk fültanúi:
```

```
-  Főnök, melyik kapcsolót kell felkapcsolni?
-  A négyest!
```

És a beosztott a főnök által megadott értéknek megfelelő folyamatot hajtja végre. Ez az „eset” talán jól példázza a C++ **switch/**Python **match** vezérlőszerkezetét!

Nézzük:

