

Ciklusok használata

Kormányos Andor

Komplex Rendszerek Fizikája Tanszék

2022. szeptember 20.

Ha a programnak többször meg kell ismételnie néhány utasítást, akkor ciklusokat írunk.

A C nyelvben összesen háromféle ciklus van

- `while(i < n) { }` – **előtesztelős ciklus**, zárójelben az ismétlés feltétele
 - a feltétel egy kifejezés, ugyanazok igazak rá, mint az if esetében
 - a feltétel a ciklusmag lefutása **előtt** értékelődik ki
- `do { } while (i < n);` – **hátraltesztelős ciklus**
 - a feltétel a ciklusmag lefutása **után** értékelődik ki
 - a ciklusmag egyszer mindenképpen lefut!
 - figyelem! a végére kell ;
- `for (...)` { } – iteratív ciklus

Nagyon gyakori, hogy egy ciklusváltozót lépésenként kell növelni (csökkenteni)

```
1 int i = 0;
2 while (i < 100) {
3     // do something
4     i++;
5 }
```

Az ennek megfelelő **for** ciklus (csak C99 szabvány!)

- két sorral tömörebb
- a ciklusváltozót nem tudjuk véletlenül a cikluson kívül használni

```
1 for (int i = 0; i < 100; i++) {
2     // do something
3 }
```

A `for` ciklus részletes szintaktikája

```
1  for ( [init] ; [condition] ; [increment] ) {  
2    // do something  
3  }
```

A két zárójelen belüli pontosvessző kötelező

Zárójelen belüli tagok:

- 1 **init**: utasítás, ami az első iteráció előtt hajtódik végre
 - jellemzően a ciklusváltozó inicializálása
- 2 **condition**: kifejezés, minden iteráció előtt kiértékelődik
 - ha értéke $\neq 0$, akkor a ciklusmag lefut
 - előltesztelési ciklus, mint a **while**
- 3 **increment**: utasítás, ami a ciklusmag legvégén hajtódik végre
 - tipikusan a ciklusváltozó növelésére, csökkentésére
 - csak akkor fut le, ha a feltétel teljesült, és a ciklusmag is lefutott

Néhány példa `for` ciklusra

Ciklusok egymásba ágyazhatóak

```
1 for (int i = 0; i < 100; i++) {  
2     for (int j = 0; j < 20; j++) {  
3         // do something  
4     }  
5 }
```

Lefelé és léptethetjük a ciklusváltozót

```
1 for (int i = 99; i >= 0; i--) {  
2     // do something  
3 }
```

Figyeljünk oda:

- C nyelvben a ciklusokat sokszor 0-tól $n - 1$ -ig futtatjuk
- ez a memóriacímzés miatt van így

A `break` és a `continue` utasítás

A `break` utasítás kilép a (legbelső) ciklusból

- általában valamilyen feltétel teljesülése esetén
- nem iteráció végén, hanem speciális esetekben használjuk
- `for` ciklusnál is működik, de többnyire `while`-nál

A `continue` utasítás átugorja a ciklusmag hátralevő részét

- ezt is feltétellel együtt használjuk
- a ciklus maga folytatódik, csak a ciklusmag nem fut tovább
- nem az iteráció végén, hanem speciális esetekben használjuk

- Írjunk egy Fibonacci-sorozatot előállító programot **for** ciklusra és hátultesztelős **do - while** ciklusra!
- Készítsünk egy programot, ami kilistázza a püthagoraszi számhármásokat, tehát az $a^2 + b^2 = c^2$ feltételnek eleget tevő a, b, c pozitív egész számhármásokat