

13. Ismétlődő tevékenységek: feltételes ciklusok

A feltételes ciklus

Ciklusokkal már találkoztunk. A hozzájuk tartozó utasítások ismétlését írták elő. A számlálós ciklus ciklusváltozó segítségével határozza meg a szükséges ismétlések számát. A ciklus kezdetekor pontosan ismert a ciklusváltozó kezdő- és végértéke (továbbá a lépésköz).

Nem mindig tudjuk azonban előre megmondani, hányszor hajtsuk végre a ciklusmagot. Lehet, hogy a cikluson belül derül ki, szükség van-e további ismétlésre.

Feltételes ciklus: az ismétlődés egy logikai kifejezés értékétől függ.

Addig üsd a vasat, amíg meleg! – tartja a közmondás. Feltételes ciklust fogalmazzunk meg. Nem tudjuk előre, hogy meddig fogjuk ütni. Az ismétlések száma a feltétel teljesülésétől függ.

Addig jár a korszó a kútra, amíg el nem törik. Ebben a ciklusban egy feltétel tagadását találjuk. Ha eltörik a korszó, akkor már nem jár többé a kútra. A kilépés feltétele, hogy eltörik a korszó. Amíg ez nem teljesül, addig ismétlődik a ciklusmag.

Ismétlési feltétel: igaz értéke a ciklusmag végrehajtásához vezet.

Kilépési feltétel: igaz értéke esetén befejeződik a ciklus. Ismétlésre a hamis érték esetén kerül sor.

A mondatszerű leírásban az „amíg” és a „míg nem” kötőszavakkal különböztetjük meg a kétféle feltételt. Az „amíg” esetén a logikai kifejezés igaz értéke, a „míg nem” használatakor pedig a hamis értéke eredményezi a ciklusmag ismételt végrehajtását.

CIKLUS AMÍG *ismétlési feltétel*
' ismétlés, amíg a feltétel igaz
utasítások
CIKLUS VÉGE

CIKLUS MÍGNEM *kilépési feltétel*
' ismétlés, amíg a feltétel hamis
utasítások
CIKLUS VÉGE

Ciklus ismétlési és kilépési feltétellel

Kérjünk be például a felhasználótól egy számot, és határozzuk meg azt a legkisebb hatványát, amely már nagyobb 10 000-nél:

Be: Szám

Hatvány = Szám

CIKLUS AMÍG Hatvány \leq 10000

Hatvány = Hatvány * Szám

CIKLUS VÉGE

Ki: Hatvány

Be: Szám

Hatvány = Szám

CIKLUS MÍGNEM Hatvány > 10000

Hatvány = Hatvány * Szám

CIKLUS VÉGE

Ki: Hatvány

I. A programozás alapjai

Vegyük észre, hogy a kilépési feltétel éppen az ismétlési feltétel tagadása. Néhány programozási nyelvnél nem is található meg mindkét típus.



Programozási összefoglaló: Feltételes ciklus



1. gyakorlat. Készítsük el a 10 000-nél nagyobb hatványt meghatározó programot!

Elöl- és hátulatesztelő ciklusok

A feltétel kiértékelésének helyétől függően a ciklusokat két csoportba oszthatjuk.

Elöltesztelő ciklus: a feltételt a ciklusmag végrehajtása előtt értékeljük ki.
Hátulatesztelő ciklus: a feltételt a ciklusmag végrehajtása után értékeljük ki.

Az előltesztelő ciklusnál előfordulhat, hogy a ciklusmag utasításai egyszer sem kerülnek végrehajtásra. A hátulatesztelő ciklus egyszer mindenképpen lefut. (A számlálós ciklusok általában az előltesztelő csoportba tartoznak.)

CIKLUS

utasítások

' ismétlés, amíg a feltétel igaz

AMÍG *ismétlési feltétel*

CIKLUS VÉGE

CIKLUS

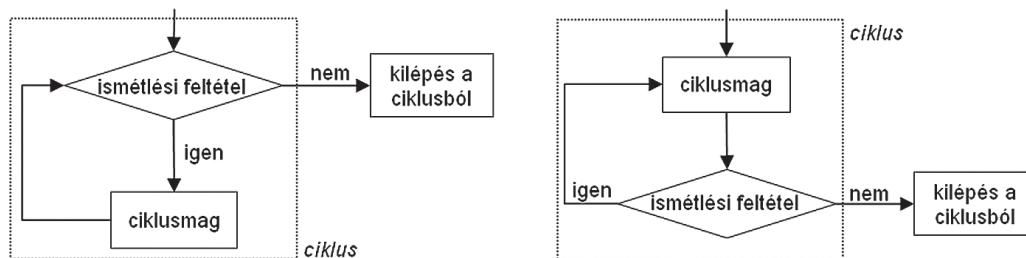
utasítások

' ismétlés, amíg a feltétel hamis

MÍGNEM *kilépési feltétel*

CIKLUS VÉGE

Hátulatesztelő ciklusok



Elöl- és hátulatesztelő ciklus folyamatábrája ismétlési feltétellel

Nem minden programozási nyelv rendelkezik a feltételes ciklusok mind a négyféle típusával (elől- vagy hátulatesztelő, ismétlési vagy kilépési feltétel).



2. gyakorlat. Alakítsuk át az 1. gyakorlat ciklusát hátulatesztelőre, de a feltételt hagyjuk változatlanul! Futtassuk a programot különböző számokra. Mit tapasztalunk?

A program tervezésekor alaposan meg kell gondolnunk, hogy elől- vagy hátulatesztelő ciklust használjunk-e, és ismétlési vagy kilépési feltételt adjunk-e meg.

Törekedjünk minél egyszerűbb, világosabb szerkezet létrehozására! Ha például az ismétlési feltételben tagadás szerepel, akkor lehetőség szerint írjunk helyette kilépési feltételt! A korsóra vonatkozó közmondást inkább így fejeznék ki: Addig jár a korsó a kútra, míg nem eltörik.

A feltételes ciklus alkalmazása

Ciklusok segítségével rejtvényeket, fejtörőket is megoldhatunk. Az esetenként bonyolult vagy nem is létező matematikai eljárás helyett számítógéppel végigpróbáljuk a lehetőségeket. Ez a módszer főleg akkor alkalmazható, ha természetes számokat keresünk.

Diophantosz ógörög matematikus sírkövére például a hagyomány szerint a következő verset vésték:

*Ím ezen emlékmű Diophantosz hamva fölött áll,
Élte korát adják művei és ez a kő.
Ifjúként tölté hatod éltét isteni kegyből,
Még tizenketted után, gyenge szakállá kiüt.
Egy heted élttel utóbb nászfáklyák égtek előtte,
Öt évvel azután kislia megszületett.
Ó, boldogság nélküli ifjú, fél annyi évet
Mint a tudós apa élt, s lett hona szürke Hádész.
Még négy esztendőn a tudás könnyíti a gyászát,
S hosszú élet után őt is elérte a vég.¹⁶*



Diophantosz görög matematikus (kb. i. sz. 210–280)

A megoldást persze némi számolással könnyen megkaphatjuk, de az alábbi ciklussal is célhoz érünk:

VÁLTOZÓ Kor MINT Valós = 1 ' Az osztás miatt valós típusú változót használunk!

CIKLUS MÍGNEM Kor/6+Kor/12+Kor/7+5+Kor/2+4=Kor

Kor+=1

CIKLUS VÉGE

Ki: Kor



3. gyakorlat. Készítsük el a Diophantosz életkorát meghatározó programot!

A végtelen ciklus elkerülése

A ciklus meghatározza a matematikus korát, de egy nagyon veszélyes lehetőséget rejt magában. Ha a feladatnak nincs megoldása, akkor sosem ér véget. A feltételes ciklusoknál különösen ügyelnünk kell a végtelen hurok elkerülésére. **Bővítsük a feltételt**

¹⁶ Egmont Colerus: Az egyszeregytől az integrálig (Franklin Társulat, Budapest, 128. old., a fordító és az évszám megjelölése nélkül)

egy olyan kifejezéssel, amely mindenképpen biztosítja az ismétlődések befejezését!

Az életkor minden bizonnyal kisebb, mint mondjuk 150 év, így a helyes megoldás:

VÁLTOZÓ Kor MINT Valós=1

CIKLUS MÍGNEM $Kor/6 + Kor/12 + Kor/7 + 5 + Kor/2 + 4 = Kor$ VAGY $Kor > 150$

Kor += 1

CIKLUS VÉGE

Gondoljuk meg, miért használtunk VAGY-műveletet!



4. gyakorlat. Bővítsük ki a 3. gyakorlat ciklusában szereplő feltételt a kor vizsgálatával!

Összetett feltételek

Az összetett kilépési és ismétlési feltételek átalakításánál gyakran alkalmazzuk a de Morgan-azonosságokat. Ha például a választott programozási nyelvben nem használhatunk előtesztelő ciklust kilépési feltétellel, akkor a ciklusfejet a következőképpen írhatjuk fel:

CIKLUS AMÍG NEM $(Kor/6 + Kor/12 + Kor/7 + 5 + Kor/2 + 4 = Kor$ VAGY $Kor > 150)$

A de Morgan-azonosságok alapján:

CIKLUS AMÍG $Kor/6 + Kor/12 + Kor/7 + 5 + Kor/2 + 4 \neq Kor$ ÉS $Kor \leq 150$

A bonyolult ismétlési/kilépési feltételeket nehéz áttekinteni. Ne a ciklusfejben végezzünk összetett műveleteket, inkább használjunk segédváltozót! Mivel a kifejezés értékét a ciklus előtt még nem ismerjük, hátulatesztelő ciklust alkalmazunk:

VÁLTOZÓ Kor MINT Valós=0

VÁLTOZÓ Összeg MINT Valós

CIKLUS

Kor += 1

Összeg = $Kor/6 + Kor/12 + Kor/7 + 5 + Kor/2 + 4$

MÍGNEM $Összeg = Kor$ VAGY $Kor > 150$

CIKLUS VÉGE

Miért írtuk át a Kor kezdőértékét 0-ra?



5. gyakorlat. A fenti példának megfelelően alakítsuk át a 4. gyakorlat programját! Mi történik, ha valós helyett egész típusúként deklaráljuk a változókat?

Az összetett feltételek megfogalmazásánál nagyon figyeljünk a megfelelő logikai művelet alkalmazására! Könnyű összecserélni az ÉS-műveletet a VAGY-gyal.

Ha a végtelen ciklus elkerülése a cél, akkor a kilépési feltételben általában VAGY-művelet áll. Befejeződik a ciklus, ha találtunk megoldást VAGY elértünk egy határt, amit nem akarunk túllépni. Az ismétlési feltétel ÉS-műveletet tartalmaz. Megismételjük a ciklust, ha nem találtunk megoldást ÉS még nem értük el a határt.



Kiegészítések a tankönyvhöz: Folyamatjelző az állapotsorban. Az ablak frissítése