

Java

h:\PoFi\2018\prg_elmélet\Programozasi_tetelek.pdf
Programozási tétélek

Programozási tételek

A programozási feladatok különböző feladatosztályokba sorolhatók a feladat jellege szerint.

A különböző feladatosztályokhoz tartozó megoldó algoritmus-szabályt programozási tételnek nevezzük.



A feladatosztályokat négy típusra osztjuk:

egy sorozathoz egy értéket rendelő feladatok,

egy sorozathoz egy sorozatot rendelő feladatok,

egy sorozathoz több sorozatot rendelő feladatok,

több sorozathoz egy sorozatot rendelő feladatok.



Sorozatszámítás (összegzés)

Egy sorozathoz egy értéket rendelő feladattípusokhoz.

Bemenet:

elemtípus: $t[]$

egész: N , // $t.hossza$ - elemszáma

művelet // $t[]$ típusain értelmezett

Kimenet:

elemtípus: s // végeredmény



Sorozatszámítás (N, t, s)

s = kezdőérték;

ciklus i = 0-tól N-ig i++

s = s művelet t[i];

ciklus vége

Eljárás vége



- Adjunk össze N darab szomszédos egész számot!
- Szorozzunk össze N darab szomszédos egész számot!
- Határozzuk meg K darab egész szám négyzetének összegét, és szorzatát!
- Határozzuk meg N darab szám átlagát!
- N hónapon át törlesztettünk egy kölcsönt, havonként különböző összeget is fizethettünk. Adjuk meg az eddig kifizetett összeget!
- Adott egy tekéző sorozata (fordulónként hány fát ütött), a versenyző összesített eredménye?

Megszámolás (darabszám)

Bemenet:

elemtípus: $t[]$

egész: N , // $t.hossza$ - elemszáma

T tulajdonság // $t[]$ típusain értelmezett

Kimenet:

egész: db // a T tulajdonságú elemek db

Megszámolás (N, t, db)

db = 0;

ciklus i=0-tól N-ig

 ha $t[i]$ T tulajdonságú, akkor

 db = db + 1;

 ha vége

ciklus vége

Eljárás vége



- Időmérési futóverseny alapján állapítsuk meg, hogy a diákok hány százaléka teljesítette az 5-ös szintet!
- Határozzuk meg az $A[N]$ vektor negatív elemeinek számát!
- Számoljuk meg, hogy hányszor fordul elő egy szövegben az „a” névelő!
- Határozzuk meg az $A[N]$ vektor azon pozitív elemeinek számát, amelyek közvetlenül egy negatív elem után állnak!
- Állapítsuk meg, hogy az $A(N)$ vektorban negatív vagy pozitív számból van-e több!



Elől tesztelő ciklus

A **while** ciklus: utasításblokk végrehajtására használható, **amíg a feltétel igaz**.

```
while (feltétel) {  
    utasítások  
}
```

A ciklusba való belépés előtt belépési feltétel igaz-e, és ha ez teljesül, akkor a ciklusmag végrehajtásra kerül, egyébként nem.

Ennél a ciklus fajtánál kialakulhat üres ciklus, abban az esetben, ha a belépési feltétel soha nem teljesül.



Bemenet:

elemtípus: $t[]$

egész: N , // $t.hossza$ - elemszáma

T tulajdonság // $t[]$ típusain értelmezett

Kimenet:

logikai: VAN // igaz/hamis - végeredmény



Eldöntés (N, t, VAN)

$i = 0;$

ciklus amíg $i < N$ és $t[i]$ nem T tulajdonságú

$i = i + 1;$

ciklus vége

$VAN = (i < N);$ // igaz/hamis

Eljárás vége



- Egy kosárlabda csapatnak van-e 210 cm-nél nagyobb játékos?
- Egy osztály tanulói év végi osztályzatai ismeretében eldönteni, hogy van-e bukott tanulója az osztálynak!
- Egy osztály nyilvántartásából megállapítani, hogy van-e év vesztes az osztályban!
- Egy személyi nyilvántartásából kideríti, hogy dolgozik-e férfi ezen a munkaterületen!
- Ismert N autó fogyasztása. Minden autó 10 liter alatt fogyasztott-e?

