

Feladat

Valósítsa meg az egész számokat tartalmazó zsák típust! Ábrázolja a zsák elemeit (az előfordulás számukkal együtt) egy sorozatban Implementálja a szokásos műveleteket (elem betétele, kivétele, üres-e a halmaz, egy elem hányszor van a zsákban), valamint két zsák különbségét (a közös elemek csak akkor maradnak meg, ha a kisebbítendő zsákban az előfordulási számuk nagyobb, mint a kivonandó zsákban, és ekkor az új előfordulási számuk a különbség lesz), továbbá egy zsák kiírását, és végül a másoló konstruktort és az értékadás operátort!

Zsák típus

A feladat lényege egy felhasználói típusnak a zsák típusnak a megvalósítása.

Típusérték-halmaz

Olyan számokat (ebben az esetben egész számokat: \mathbb{Z}) tartalmazó n -es ($n \in \mathbb{N}$) multihalmazokkal akarunk dolgozni. Multihalmaz olyan halmaz, melyben egy elem többször is előfordulhat. Az elemek sorrendje nem számít.

Típus-műveletek

1. Elem betétele

A zsákba e elemből m darab betétele ($e \in \mathbb{Z}$, $m \in \mathbb{N}^+$).

2. Elem kivétele

A zsákban lévő e elemből m darab kivétele. ($e \in \mathbb{Z}$, $m \in \mathbb{N}^+$).

Megjegyzés: Ha e elem nem eleme a zsáknak, vagy e elemből m -nél kevesebb elem van a zsákban, hibás.

3. Üres-e a halmaz

Adott zsákról megállapítás, hogy van-e eleme vagy nincs.

4. Elem előfordulása egy zsákban

Egy adott e elemről megadja, hogy hányszor szerepel a zsákban.

Megjegyzés: Ha e nem eleme a zsáknak, az előfordulási száma nulla.

5. Két zsák különbsége

Két zsák különbsége $c := a - b$. A nem közös elemekből a kisebbítendő (a) összes eleme megmarad, a közös elemek csak akkor maradnak meg, ha a kisebbítendőben (a) többször fordulnak elő, ekkor a két zsákban való előfordulásuk különbsége lesz az eredményben való mennyiségük.

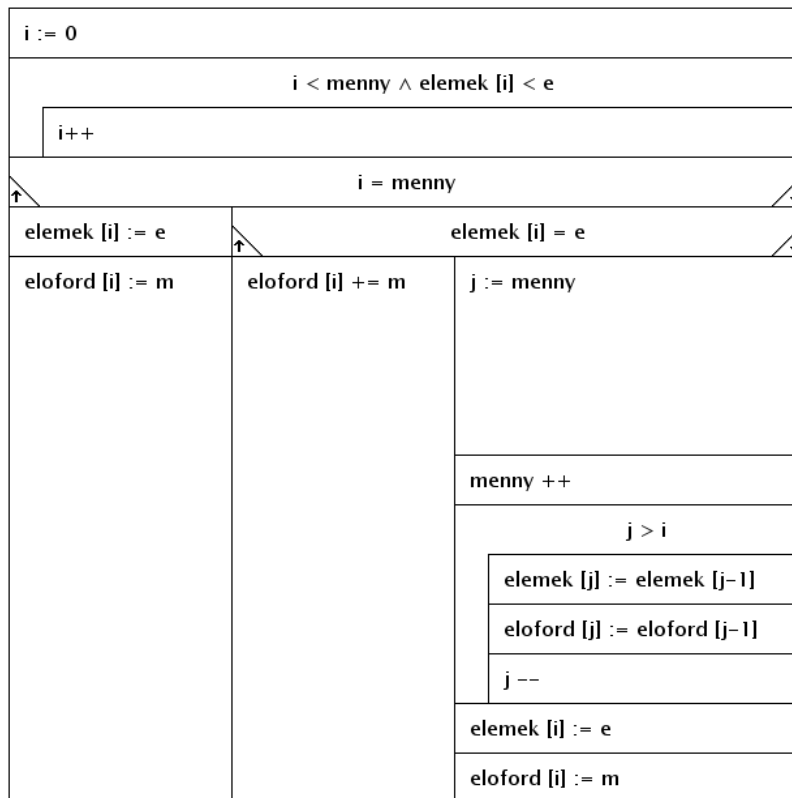
Reprezentáció

A zsákot két dinamikusan lefoglalt egydimenziós tömbbel, és a különböző elemek mennyiségével (*menny*) reprezentáljuk. Az egyik tömb ekkor a zsákban lévő elemek rendezve, ismétlődés nélkül (*elemek*), a másik tömb pedig az adott indexű elem mennyiségének a száma (*előford*).

Implementáció

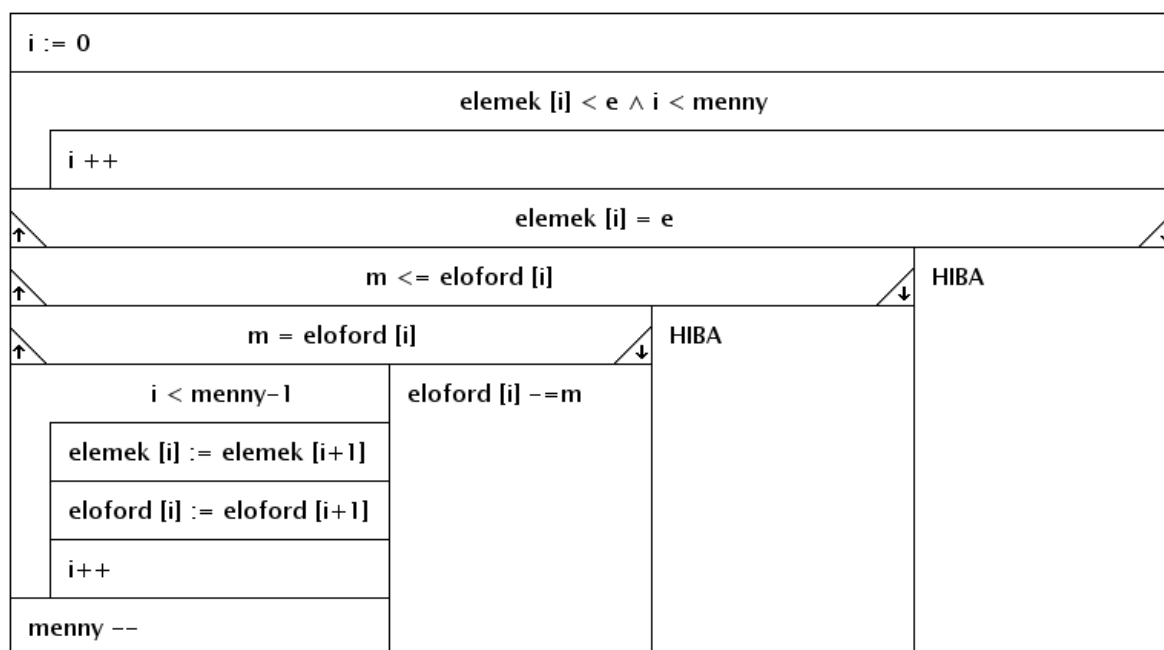
1. Elem betétele

Az *elemek* és *eloford* tömbbel és *menny* számmal ábrázolt zs zsákba való *e* elem *m*-szer való betétele az alábbi programmal implementálható, feltéve, hogy $e \in \mathbb{Z}$ és $m \in \mathbb{N}^+$:



2. Elem kivétele

Az *elemek* és *eloford* tömbbel és *menny* számmal ábrázolt zs zsákból az *e* elemből *m* darab kivétele az alábbi programmal implementálható, feltéve, hogy $e \in zs$ és $m \in \mathbb{N}^+$, továbbá *e* elem legfeljebb *m*-szer fordul elő zs-ben:



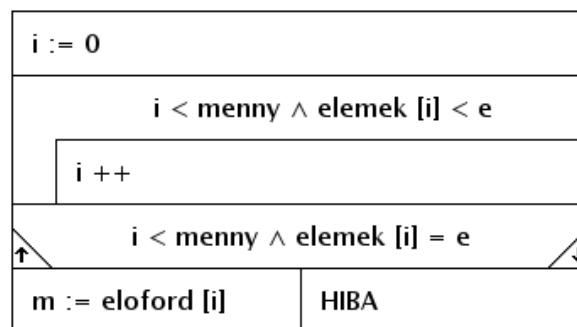
3. Üres-e a halmaz

Az *elemek* és *eloford* tömbbel és *menny* számmal ábrázolt zs zsákról a következő programmal implementálható, hogy *l* logikai érték, amely akkor igaz, ha zs üres zsák.:

$l = \text{menny} == 0$

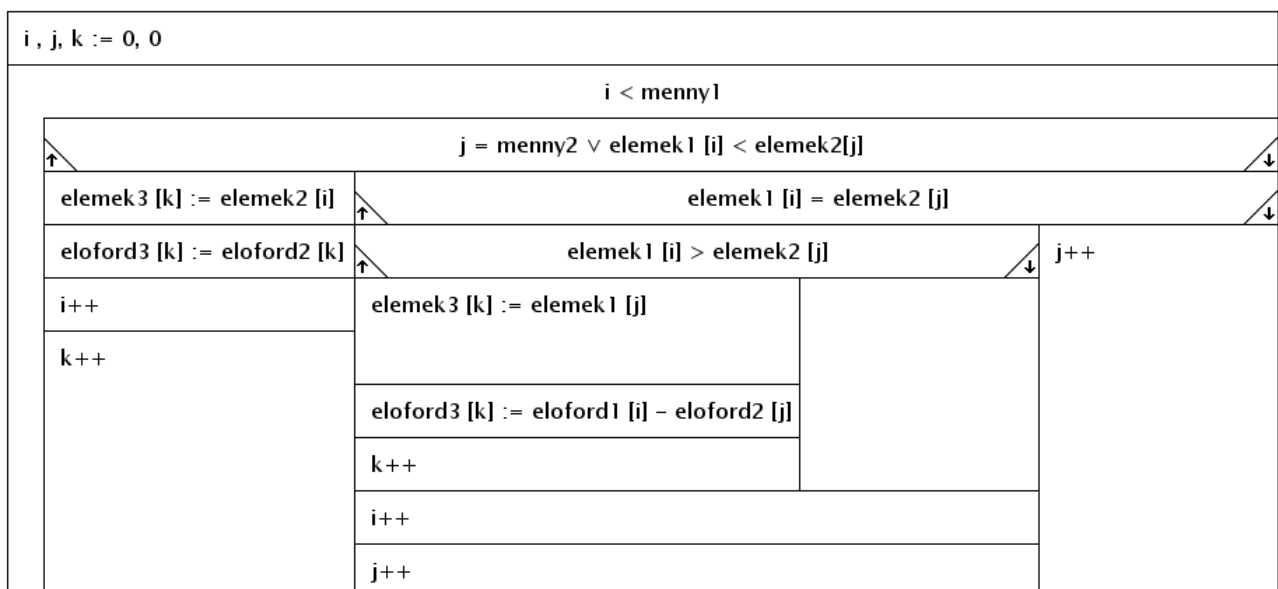
4. Elem előfordulása egy zsákban

Az *elemek* és *eloford* tömbbel és *menny* számmal ábrázolt zs zsákban adott *e* elem előfordulásának száma a zsákban a következő programmal implementálható, ahol *e* elem *m*-szer fordul elő zs zsákban.



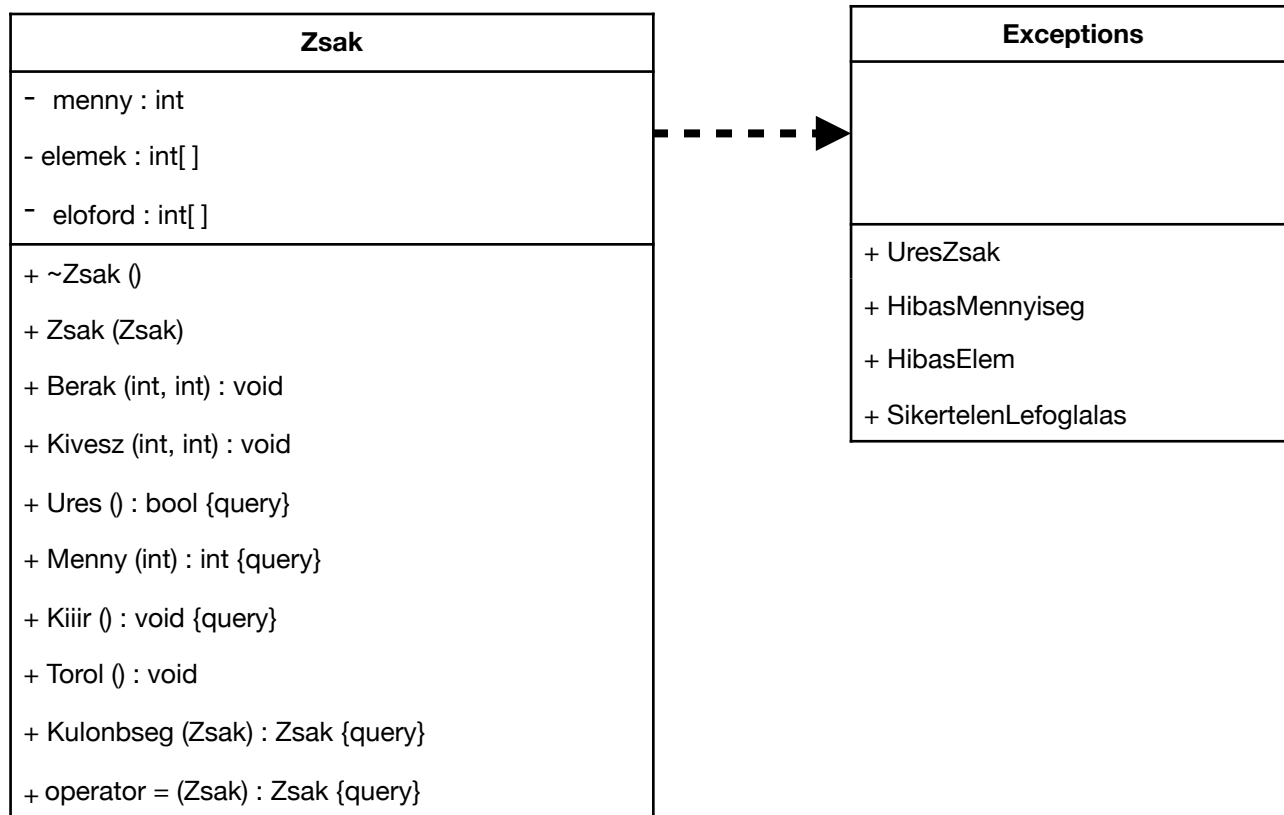
5. Két zsák különbsége

Az *elemek1* és *eloford1* tömbbel és *menny1* számmal ábrázolt *zs1* zsák, és az *elemek2* és *eloford2* tömbbel és *menny2* számmal ábrázolt *zs2* zsák különbsége ($zs3 := zs1 - zs2$) az alábbi programmal reprezentálható, ahol *zs3* a két zsák különbsége, amely *elemek3* és *eloford3* dinamikus tömbökkel, illetve *menny3* számmal ábrázolható:



Osztály

A zsákok típusát egy osztály segítségével valósítjuk. Egyes műveleteknél maj ellenőrizni kell, hogy a mátrix üres-e (elem kivétele).



A függvények az adott típus-műveletek megvalósítására szolgálnak.

UresZsak hibaüzenetet akkor kapunk, ha üres zsákból szeretnénk elemet kivenni, *HibasMeret*-et akkor, ha nem megfelelő darabszámú elemet szeretnénk a zsákba rakni vagy kivenni (pl: negatív darabot vagy többet, mint ahány benne van). *HibasElem* hibaüzenetet kapunk, ha a *Kivesz* műveletnél nincs olyan elem a zsákban, mint ami ki szeretnénk venni, *SikertelenLefoglalas* pedig a zsák konstruktor sikertelen lefutása esetén adódik.

Tesztelési terv

Megvalósított műveletek tesztelése (fekete doboz tesztelés)

- 1) Konstruktorok
 - a) Zsák létrehozása, ürességének vizsgálata
 - b) Másoló konstruktor kipróbálása
 - i) A $zs2$ zsák létrehozása $zs1$ mintájára, majd $zs2$ megváltoztatása és a két zsák tartalmának összehasonlítása
- 2) Berakás művelet
 - a) Negatív mennyiségű elem berakása
 - b) Nulla darabszámú elem berakása
 - c) Elem berakásának helyessége
 - d) Elem mennyiségének berakásának helyessége
 - e) Berakáskor rendezés helyessége
- 3) Kivétel
 - a) Üres zsákból elem kivétele
 - b) Elem kivételének helyessége
 - c) Nem létező elem kivétele
 - d) Túl nagy mennyiségű elem kivétele
- 4) Üres-e
- 5) Elem előfordulása
 - a) Létező elem előfordulása
 - b) Nem létező elem előfordulása
- 6) Két zsák különbsége ($zs3 := zs1 - zs2$)
 - a) $zs1 > zs2$ vizsgálat (eredmény nem üres zsák)
 - b) $zs1 < zs2$ vizsgálata (eredmény üres zsák)
- 7) Értékadás operátor
 - a) A $zs2 = zs1$ értékadás végrehajtása a $zs1$ és $zs2$ zsákokra, majd a két zsák tartalmának összehasonlítása, majd a $zs1$ zsák megváltoztatása és a két zsák tartalmának összehasonlítása.
 - b) A $zs3 = zs2 = zs1$ értékadás végrehajtása a $zs1$, $zs2$, $zs3$ zsákokra, majd a két mártix tartalmának összehasonlítása, majd az egyik zsák megváltoztatása és a mártixok tartalmának összehasonlítása.
 - c) A $zs = zs$ értékadás végrehajtása a zs zsákra, majd ennek vizsgálata.

Tesztesetek kód alapján (fehér doboz tesztelés)

1. Hibaüzenetek generálása és elkapása