# **Zsák dokumentáció “9. feladat”**

Készítette: Tóth Zalán  
Neptun-azonosító: cz72ym  
E-mail: [zlaval@gmail.com](mailto:zlaval@gmail.com)  
  
Kurzuskód: IP-18eOEPROGEG  
Gyakorlatvezető: Dr. Gregorics Tibor  
Csoport: 1  
Secret: 16039

1. beadandó

**2020. február 26.**

**Feladat**

*Valósítsa meg az egész számokat tartalmazó zsák típust! Ábrázolja a zsák elemeit (az előfordulás számukkal együtt) egy sorozatban! Implementálja a szokásos műveleteket (adott elem betevése, kivétele, egy adott elem hányszor van benne a zsákban), valamint a zsák azon elemeinek darabszámát visszaadó műveletet, amelyek csak egyszeresen vannak a zsákban (ehhez célszerű a zsákot reprezentáló sorozat mellett külön eltárolni az egyszeres gyakoriságú elemek darabszámát, és ezt karbantartani), továbbá egy zsák kiírását!*

**Zsák típus**

A feladat egy felhasználói típusnak, a Zsák (Bag) típusnak a megvalósítása.

Típus értékhalmaz

A Zsák típus ebben az esetben egy sorozat (*v*∈*H*), mely tárolja az *e*∈ℤ sorozatelem hányszor került a Zsákba (*c*∈*ℕ*).

Típus-műveletek

1. *Elem berakása (put)*Természetes szám behelyezése a Zsákba.
2. *Elem eltávolítása (remove)*Természetes szám eltávolítása a Zsákból. Lehetőség van adott *i* elemből *a*∈*[0..ce]* darab egy művelettel történő eltávolítására.
3. *Adott elem számossága (count)  
   e*∈ℤ természetes szám ce számosságát adja meg (hányszor tettük be *e*-t a Zsákba).
4. *Egyszeres számosságú elemek (cardinality)*Visszaadja, hány olyan *e*∈ℤ elem van a Zsákban, ahol *ce=1*.
5. *Zsák mérete (size)*Hány *e*∈ℤ elem számosságát tartjuk nyilván a Zsákban. Eredménye *n:=|v|.*

Reprezentáció

A Zsák egy sorozatban tárolja az *e*∈ℤ elemeket és számosságukat (*c*).

Az adatokat egy adatosztályban tartjuk nyilván, ezek lesznek a zsák elemei.

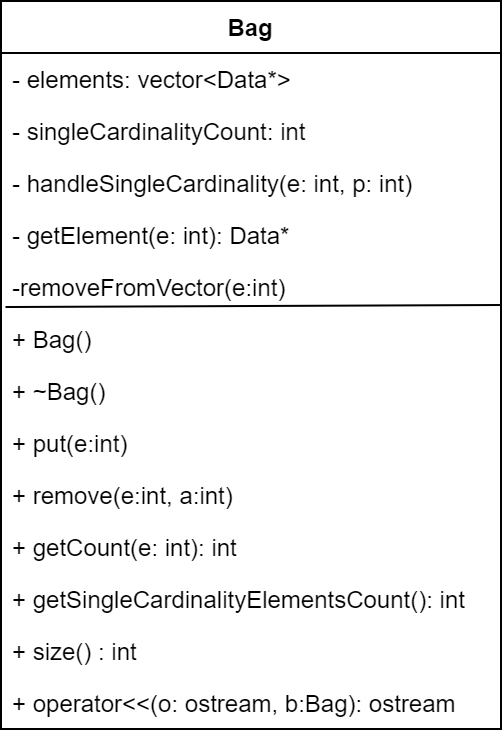


Példa: Egy 10 elemméretű Zsák reprezentációja (*i*=*index*).

i=0 i=1 i=2

|  |  |  |
| --- | --- | --- |
| Data(3,2) | Data(10,1) | Data(-1,120) |

Osztálydiagram

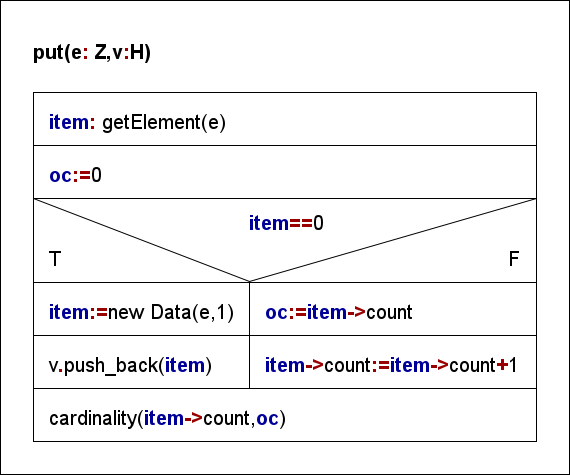


Implementáció

*1, Elem berakása*

Az *e*∈ℤ elem elhelyezésekor a Zsákban a reprezentáló *v*∈*H* vektor végére fűzzük az elemet reprezentáló osztály egy új példányát, amennyiben *e*∈ℤ nem volt eleme a Zsáknak. Ha eleme volt, lineáris kereséssel megkeressük az elemet (külön ábrán bemutatva) és eggyel megnöveljük a számosságát (ce).

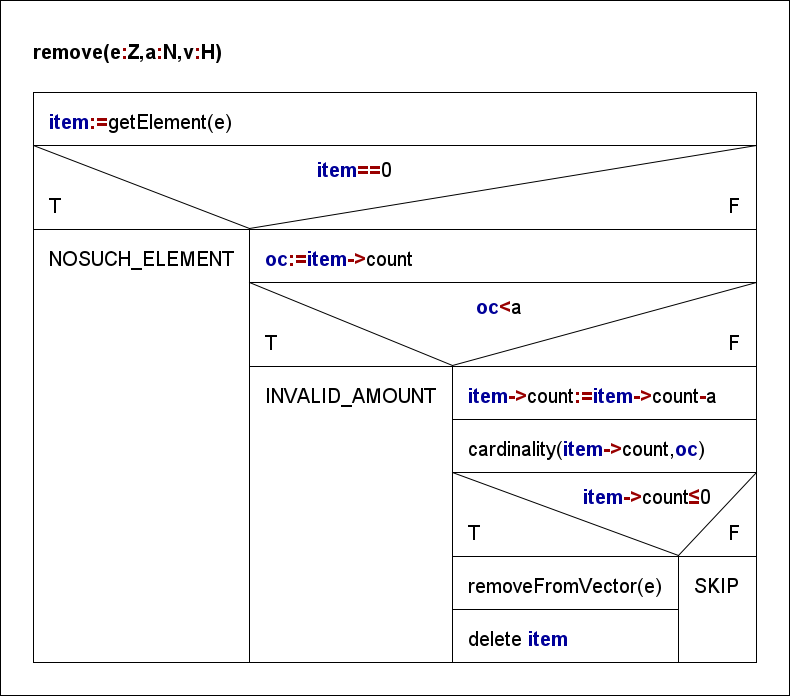
Elem behelyezésekor történik az egyszeres számosságú elemek számának karbantartása is, ez külön ábrán kerül bemutatásra az implementációs részletek végén, itt csak hivatkozás történik rá.



*2, Elem eltávolítása*

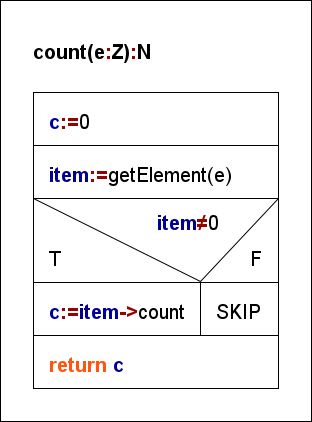
Az *e*∈ℤ elem *a*∈*[0..ce]* mennyiségű eltávolításakor a Zsákban a reprezentáló *v* vektorban *a* értékével csökken az *e elem számossága*, amennyiben *a*∈ℕ<=ce és az elem szerepel a zsákban.

Ha az elem számossága 0-ra csökken, töröljük a reprezentáló vektorból (külön bemutatva).

**

*3, Adott elem számossága*

Az *e*∈ℤ elem számossága (ce) a Zsákban.

**

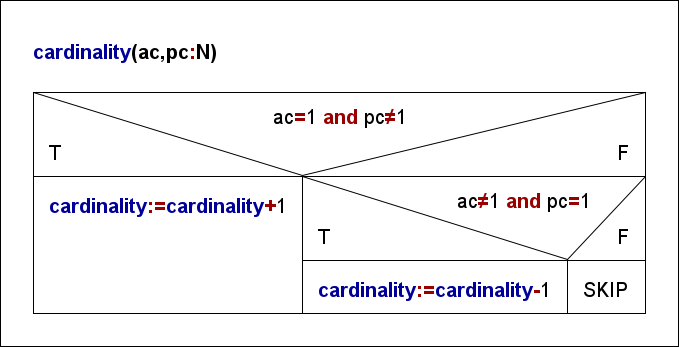
*4, Egyszeres számosságú elemek*

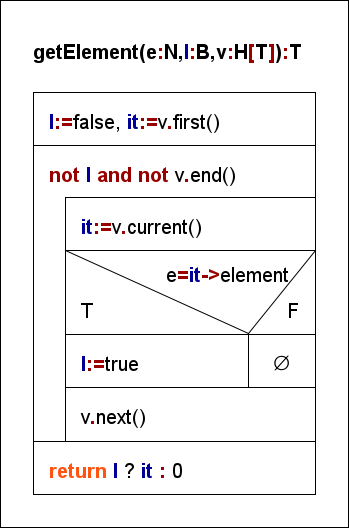
A Zsákot reprezentáló *v* vektor olyan elemeinek számát (szummáját) adja vissza, ahol ce:=1.

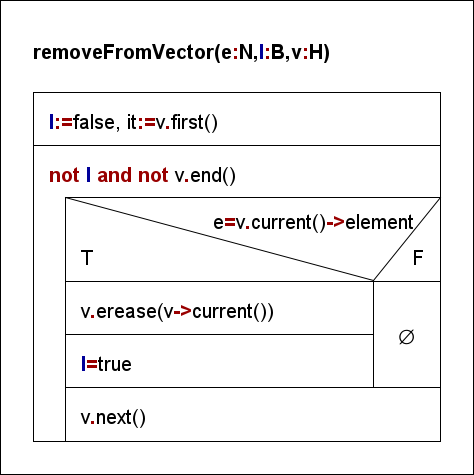
*5, Zsák méret*

A Zsákot reprezentáló *v* vektor hosszát adja eredményül.

Felhasznált segéd metódusok struktogrammjai:







**Tesztelési terv**

Megvalósított műveletek tesztjei

* **Egyszeres számosságú elemek:**
  1. Új Zsákban nincs egyszeres számosságú elem.
  2. Egy elem üres Zsákhoz hozzáadásával egy lesz az egyszeres számosságú elemek száma.
  3. Ugyanazt az elemet kétszer az üres zsákhoz adva nulla lesz az egyszeres számosságú elemek száma.
  4. Különböző elemeket a zsákba rakva az egyszeres számosságú elemek száma az egyszer behelyezett elemek számával lesz megegyező.
* **Elemek eltávolítása**
  1. Egy darab, egyszeres számosságú elemet tartalmazó zsákból az elemet eltávolítva nulla lesz az egyszeres számosságú elemek száma.
  2. Hármas számosságú elemből kettőt eltávolítva eggyel nő az egyszeres számosságú elemek száma.
  3. Kettes számosságú elemből kettőt eltávolítva nem változik az egyszeres számosságú elemek száma.
  4. Adott elem számosságánál többet törölve belőle. *INVALID\_REMOVE\_AMOUNT* kivétel keletkezik.
  5. Egyszeres számosságú elemből nulla darabot eltávolítva az elem egyszeres számosságú marad.
* **Számosság lekérdezése**
  1. A be nem helyezett elem számossága 0
  2. Üres Zsákban minden elem számossága nulla.
  3. Egy elemet N-szer a Zsákba téve a számossága N lesz.
* **Zsák méret lekérdezése**
  1. Öt elem méretű Zsák mérete öt.
* **Elem zsákba helyezése**
  1. A behelyezett elem megjelenik a zsákban

Tesztesetek kód alapján

* Terheléses teszt, e:[1..231] elemek Zsákba helyezése.