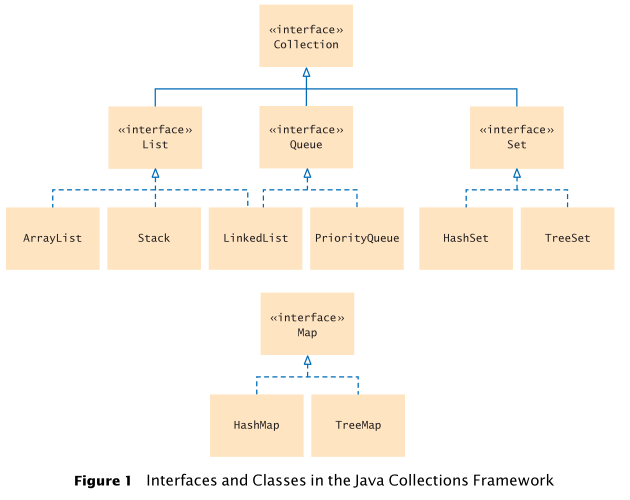
# 16. Halmazok

## Ismétlés

Ha több dolgot együtt szeretnénk kezelni egy programban, akkor azokat egy gyűjteménybe, más szóval kollekcióba (collection) tesszük. Például az ArrayList osztály egy kollekció, amelyet a Java biztosít számunkra.

A Java ezen kívül még sok más interface-t és osztályt is biztosít a különböző kollekciókhoz. Ezeket együttesen Java Collections Framework-nek nevezzük.

Az alábbi ábra mutatja a JCF főbb elemeit:  
(folytonos vonal: öröklődés, szaggatott vonal implementáció)



## Halmazok

Ebben a leckében a halmazokkal foglalkozunk. A Set (halmaz) nagyon hatékonyan tud hozzáadni, elvenni és megtalálni elemeket a kollekcióban, és nem engedi, hogy egy elem kétszer szerepeljen.

Ha olyan elemet próbálunk hozzáadni egy halmazhoz, amely már szerepel benne, nem történik meg a hozzáadás.

A Set interface-t valósítja meg a HashSet és a TreeSet osztály is.

A HashSet osztályt akkor használjuk, ha nem lényeges az elemek sorrendje. Ekkor a műveletekhez optimalizált sorrendben lesznek az elemek a halmazban.

A TreeSet osztályt pedig akkor használjuk, ha növekvő sorrendben végig kell tudni menni az elemeken. Ekkor sorba rendezve tárolódnak az elemek. Ehhez az szükséges, hogy az elemek megvalósítsák a Comparable interface‑t.

A halmazok kezeléséhez is ugyanazokat a metódusokat használhatjuk, mint a többi kollekcióhoz.

Új halmaz létrehozása:

Set<String> nevek1 = new HashSet();

Set<String> nevek2 = new TreeSet();

Új elem hozzáadása (nem lehet duplikált!):

nevek1.add("Romeo”);

A visszaadott érték true, ha hozzáadta az új elemet, false, ha nem, mert már szerepelt a halmazban.

Elem eltávolítása (ha nincs ilyen elem, nem kapunk hibaüzenetet!, de false értéket ad vissza):

nevek1.remove("Juliet");

Tartalmazás vizsgálata:

if (nevek1.contains("Romeo")) …

A contains metódus az equals metódust használja az elemek egyenlőségének viszgálatához. Ez String, Integer, … osztályok esetén adott, de ha a halmaz elemeinek típusa saját osztály, akkor abban el kell készítened az equals metódust!

A halmaz elemeinek listázásához is szükséges iterátor. Ezt a láncolt listákhoz hasonlóan használhatjuk. Például:

Iterator<String> it = nevek1.iterator();  
while (it.hasNext()) {  
 String nev = it.next();  
 // elem feldolgozása  
}

Itt is egyszerűbben használható a for-each ciklus:

for (String nev : nevek1) {  
 // utasítások  
}

Az iterátor a tárolás sorrendjében megy végig az elemeken. TreeSet esetén ez növekvő sorrend, HashSet esetén pedig „össze-vissza” (lásd a következő szakaszban).

Fontos különbség a láncolt listákhoz képest, hogy a halmazoknál közvetlenül a halmazhoz lehet hozzáadni elemet, nem az iterátorhoz! (Törölni viszont lehet az iterátor által mutatott elemet.)

Ezen kívül visszafelé sem lehet haladni az elemeken az iterátorral.

## A HashSet működése

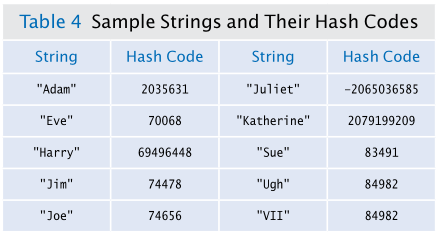
Kérdés: Hogyan csoportosítjuk egy kézzel írott telefonregiszterben a bejegyzéseket? Hogyan segíti ez egy telefonszám gyors kikeresését?

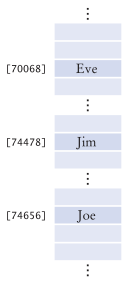
=

Az alapötlet az, hogy az elemek egy tömbbe kerülnek, egy olyan helyre, amit maga az elem határoz meg a hash kódjával.

A hash kód egy egész szám, amit úgy számol ki a program, hogy egy objektumnak mindig azonos legyen, a különböző objektumoknak pedig különböző legyen a kódja. Utóbbit nem lehet teljesen megvalósítani, de törekednek az ütközések (azonos kódok) minimalizálására.

A hash kód kiszámításának pontos módját szerencsére nem muszáj tudnunk a használatához. Az alábbi táblázat mutat néhány példát stringek hash kódjára:



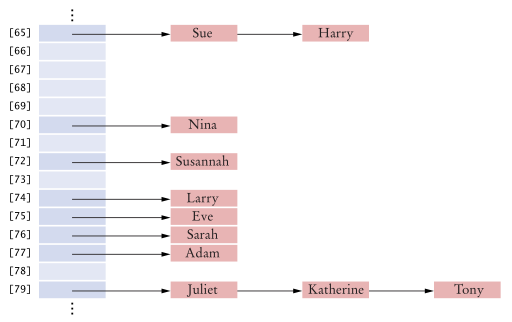
Ha minden stringet egy a hash kódjával megegyező indexű helyre tennénk, az a bal oldalon látható tömböt eredményezné.

Ha nem lenne ütközés, akkor nagyon egyszerű és gyors lenne meghatározni, hogy egy string szerepel-e már a halmazban: ki kell számolni a stringhez tartozó hash kódot, és megnézni, hogy foglalt-e már a hely. A tömb többi elemét meg sem kell nézni!

Ekkora tömböt azonban nem foglalhatunk le a halmaznak, ezért a hash kódok nagyságát korlátozni kell.

Ezt úgy lehet megoldani, hogy vesznek egy elfogadható méretű tömböt, és a hash kódokat elosztják ennek méretével, majd veszik a maradékot. (Negatív hash kód esetén először az ellentettjét veszik.)

A kódok korlátozása után gyakoribbá válik az ütközés. Ennek egyik lehetséges megoldása, hogy az azonos elembe kerülő elemekből egy listát képeznek:



Ebben az adatszerkezetben a következő lépésekkel lehet megvizsgálni, hogy egy objektum szerepel-e benne:

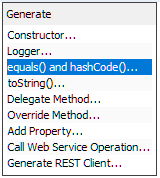
1. Meg kell határozni a hash kódját (a korlátozással).
2. Végig kell nézni a tömb hash kóddal megadott helyén lévő listát.
3. Ha az objektum szerepel a listában, akkor eleme a halmaznak, egyébként nem.

Ez még mindig sokkal gyorsabb, mint az összes elemet végignézni.

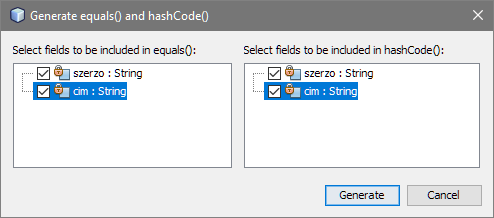
A Java könyvtárában kapott osztályok (String, Integer, …) tartalmazzák a megfelelő hashCode() metódust saját hash kódjuk előállításához.

Ha azonban saját osztály példányait tartalmazza a halmaz, akkor felül kell írnunk az Object osztálytól örökölt hashCode metódust.

Szerencsére a hashCode (és vele együtt az equals) metódusokat a NetBeans is el tudja nekünk készíteni. Ehhez az Alt+Insert billentyűket kell megnyomnunk, majd a menüből kiválasztani az *equals() and hashCode()* parancsot:



A következő ablakban meg kell adni, hogy melyik mezőket vegye figyelembe a hash kód generálásánál, illetve az egyenlőség vizsgálatánál:



Nem muszáj mindet bejelölni!

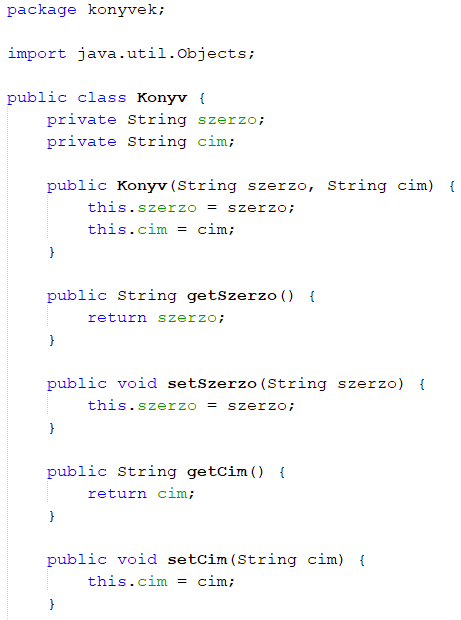
## Mintaprogram

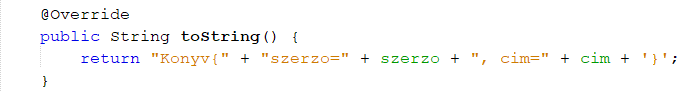
Két barát könyveket gyűjt. Szeretnék tudni, hogy melyek azok a könyvek, amelyek mindketőjüknek megvannak.

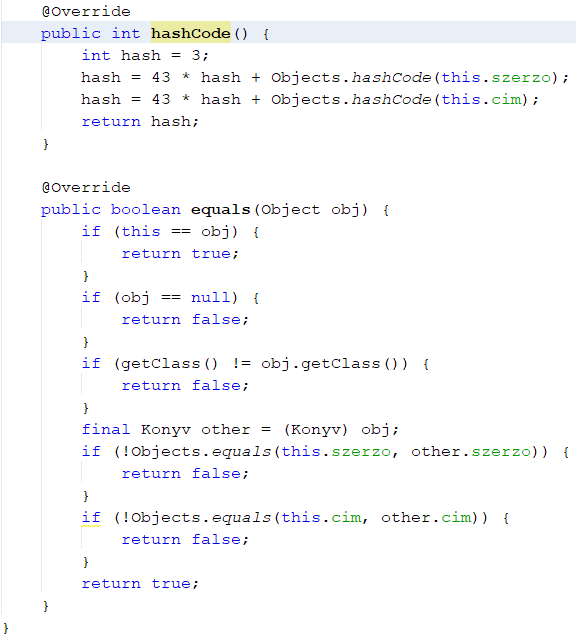
A könyveket két halmazba fogjuk tenni, és kiíratjuk azokat, amelyek mindkettőben szerepelnek.

Kezdj egy új projektet *konyvek* néven!

Hozz létre egy új osztályt Konyv néven, és készítsd el a tartalmát az alábbiak szerint! Csak a két adattagot írd be kézzel, a többit generáltasd a programmal! (Alt+Insert)

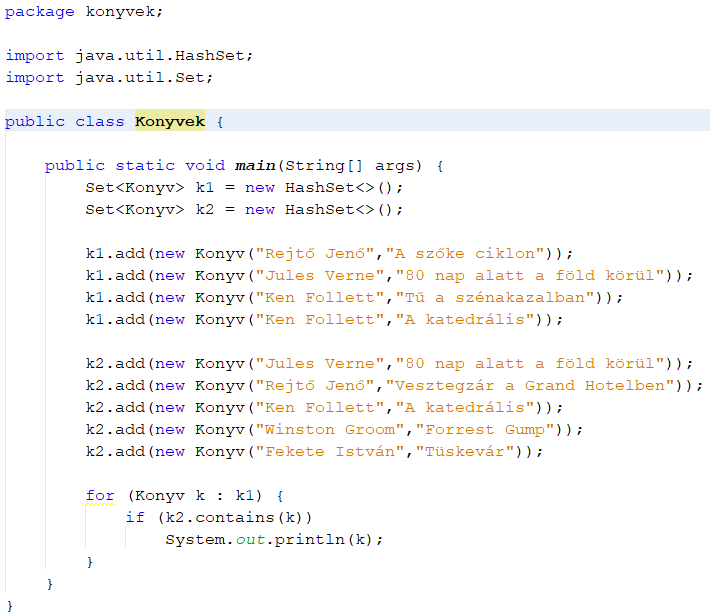






Figyeld meg a hashCode és az equals metódusokat!

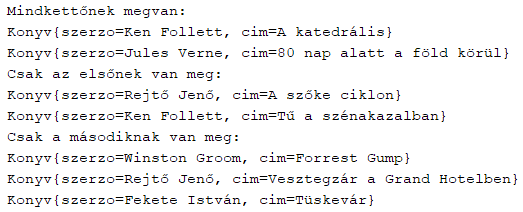
A főprogramban létrehozunk két HashSet-et, hozzáadunk néhány könyvet, majd kiíratjuk azokat a könyveket, amelyek mindkét halmazban szerepelnek:



Írd be, és próbáld ki!

Egészítsd ki a programot úgy, hogy kiírja azokat a könyveket, amelyek csak az első halmazban, majd azokat is, amelyek csak a másodikban szerepelnek!

Minta:



## Kiegészítés

A HashMap osztály HashSet-hez hasonlóan tárolja a párokat.