

10. feladatsor

Kezdés: ápr 26, 16:47

Kvízinstrukciók

1. kérdés

0 pont

Programozási nyelvek (BSc, 18) Java 10. feladatsor

1. feladat

A 8. feladatsor 1. feladatának kódját fejlesszük tovább.

Írjon a `Book` osztály `pages` adattagjához getter metódust (`getPages()`).

Készítsünk egy `Article` osztályt, amely avalemilyen cikket (pl. cikk, bírálat, ajánló) reprezentál. Ahogy a 8. feladatsorban láttuk, ilyen cikkek hivatkozhatnak könyvekre, és az ilyen hivatkozásokat gondosan össze szokták szedni egy referenciagyűjteményben. Ebben a feladatban csak egy ilyen referenciagyűjtemény fájlba írására fókuszálunk, a cikk (`Article`) egyéb logikájával nem foglalkozunk.

Egy cikknek van címe, törzse és konklúziója (`title`, `body`, `conclusion` sztringek), `PrintWriter`-e és `Book` tárolására alkalmas `ArrayList`-je (`refs`). Írjon konstruktort, amely inicializálja egy `Article` adattagjait.

Írjon `addBookToReferences()` néven metódust, amely a paraméterként kapott könyvet (ez lehet `Book`, `PrintedBook` vagy `EBook` is) hozzáadja az `Article` `refs`-jéhez.

Írjon `print()` metódust, amely egy már elkészített cikk teljes terjedelmét a paraméterként kapott fájlnevű fájlba írja. Írja ki sortöréssel elválasztva a cikk címét, törzsét, konklúzióját majd a hivatkozott könyvek hivatkozási formáit (`createReference()`). Készítsen privát segédfüggvényt, amely egy adott könyv esetén fájlba írja a könyv hivatkozási formáját, ahol a cikknév az aktuális cikk neve, a kezdőoldalszám 1, a végoldalszám a könyv utolsó oldala (`getPages()`).

Hogyan valósítanánk meg ezt a programot felüldefiniálás használata nélkül? Milyen hátrányai vannak egy ilyen megoldásnak az itt bemutatott megoldással szemben?

2. feladat

Készítsen egy `Bag<T>` generikus osztályt, mely egy zsákot valósít meg. A zsák olyan halmaz, mely többször tartalmazhatja ugyanazt az elemet.

Legyen egy `HashMap<T, Integer>` adattagja, melyet a paraméter nélküli konstruktor megfelelően feltölt. Legyen egy `add(T element)` metódusa. Ellenőrizze, hogy van-e már ilyen kulcs a zsákban, ha nincs, tegye bele `1` értékkel. Ha van, kérdezze le az aktuális

értéket, és tegye bele az `1`-gyel megnövelt értéket (azt tároljuk a map-ben, hogy melyik objektum hányszor van a zsákban).

Legyen egy egész visszatérési értékű `countOf(T element)` metódusa, mely megadja, hogy hányszor van az elem a zsákban. Ha nincs az elemhez mint kulcshoz rendelve semmilyen érték a map-ben, adjon vissza `0`-t.

Legyen a zsáknak egy `remove()` metódusa is egy elem kivételére. Csökkentse `1`-gyel a megadott elem darabszámát a zsákban. Ha `0`-ra csökken a darabszám, vegye ki a megfelelő kulcs-érték párt a map-ből, hogy ne tároljunk feleslegesen adatokat. Ha az elem nem volt a zsákban, dobjunk `NotInBagException` kivételt, amely legyen egy saját kivétel osztály. A `NotInBagException` kivétel származzon `Exception`-ből, és a sztringet fogadó konstruktora hívja meg az őssztály konstruktorát.

Készítsen főprogramot, amely a `Bag<T>` generikus típus felhasználásával beolvas egy olyan szöveges fájlt, amely soronként egy szót tartalmaz, majd a szavak előfordulási gyakoriságáról statisztikát készít.

input.txt:

```
hello
world
interface
abstract
abstract
world
world
world
world
hello
world
X-Files
protected
abstract
abstract
extends
protected
socket
world
hello
socket
extends
```

3. feladat

Írjon `swap()` néven generikus metódust, amely megcseréli a paraméterként kapott egydimenziós tömb két tetszőleges elemét.

1. gyakorló feladat

Készítsen egy `CheckedSet<T>` generikus osztályt, mely egy halmazt valósít meg. Legyen egy `HashSet<T>` adattagja, melyet a paraméter nélküli konstruktor megfelelően feltölt. Lehessen lekérdezni a halmaz aktuális méretét. Legyen egy `add(T element)` metódusa, mely dobhat `AlreadyContainedException` ellenőrzött kivételt. Ehhez hozzuk létre ezt a kivételosztályt, és dobjuk akkor, ha a halmaz már tartalmaz a megadottal egyenlő elemet;

különbön tegyük be a halmazba az új elemet. Legyen egy logikai visszatérési értékű `contains(T element)` metódusa, mely ellenőrzi, hogy a halmaz tartalmazza-e a megadott elemet. Egy főprogramban teszteljük le az új osztályt.

2. gyakorló feladat HF

Készítsünk egy `Stack<T>` generikus osztályt, amely egy vermet reprezentál (elem elhelyezése a verem tetején, elem kivétele a verem tetejéről, legfelső elem lekérdezése, méret lekérdezése, kivételek dobása hibák esetén)!

Feltöltés

Fájl kiválasztása

Kvíz mentve ekkor: 16:48

Kvíz beadása