

Programozási nyelvek II. JAVA

8. gyakorlat

2017. november 6-10.

Általános tudnivalók

- A feladatmegoldás során fontos betartani az elnevezésekre és típusokra vonatkozó megszorításokat, illetve a szövegek formázási szabályait.
- Segédfüggvények létrehozhatóak, a feladatban nem megkötött adattagok és elnevezéseik is a feladat megoldójára vannak bízva.
- Törekedjünk arra, hogy az osztályok belső reprezentációját a lehető legjobban védjük, tehát csak akkor engedjük, és csak olyan hozzáférést, amelyre a feladat felszólít, vagy amit azt osztályt használó kódrészlet megkíván!
- A feladat megoldása során be kell tartani a **kóddolási konvenciókat**.

A feladat összefoglaló leírása

- A feladatban egy piac nagyon leegyszerűsített működését fogjuk szimulálni.
- A piacon gyümölcsöket árulnak, amelyeknek nyilvántartjuk a nevüket és az árukat.
- Nyilvántartjuk, hogy a piacon milyen gyümölcsöket lehet kapni.

A feladat részletes ismertetése (gyümölcs)

- Készítsd el a `market.Fruit` osztályt, ami egy gyümölcsöt reprezentál.
- Az osztálynak két rejtett adattagja van: egy szöveges típusú `name`, amely a gyümölcs nevét tárolja és egy egész szám típusú `price`, ami a gyümölcs árát tárolja.
- Az osztálynak legyen egy rejtett konstruktora, amely paraméterben megkapja a nevet és az árat, és beállítja a megfelelő adattagokat.
- Legyen egy paraméter nélküli `getPrice`, ami visszaadja a gyümölcs árát.
- Az osztály tartalmazzon egy `cheaperThan` metódust, ami eldönti, hogy az aktuális gyümölcs olcsóbb-e, mint a paraméterben kapott gyümölcs.

A feladat részletes ismertetése (gyömolcs)

- Legyen egy statikus `make` metódus, amely szintén egy nevet és egy árat kap. A metódusnak ellenőriznie kell a paramétereket, és amennyiben azok megfelelőek, akkor hozza létre, és adja vissza a paramétereknek megfelelő `Fruit` objektumot. Ha a paraméterek nem jók, akkor a metódus `null`-t adjon vissza. A nevet tartalmazó paraméter akkor megfelelő, ha csak betűt tartalmaz és legalább 2 karakter hosszú. Az árat tartalmazó paraméter pedig akkor helyes, ha pozitív, de legfeljebb 5000, továbbá és 0-ra vagy 5-re végződik.

A feladat részletes ismertetése (gyömölcs)

- Az osztályban legyen egy paraméter nélküli `show` metódus, amely visszaadja az objektum szöveges reprezentációját. A formátum legyen a következő:
`név (ár-ezres-tagolással Ft)`
(ha az ár 1000-nél nagyobb, akkor ezres tagolásként egy szóközt kell használni, pl. `3 065`).

A feladat részletes ismertetése (gyümölcs)

- Az osztály tartalmazzon egy `cheapestFruit` nevű osztályszintű adattagot, ami a legolcsóbb gyümölcsöt tartalmazza, amit valaha létrehoztak (ha több ilyen van, akkor ezek közül az elsőt). Ha még nem hoztak létre `Fruit` objektumot, akkor az adattag értéke legyen `null`. Figyelj rá, hogy a konstruktor mindig aktualizálja ezt az objektumot, amikor az összes eddiginél olcsóbb gyümölcsöt hoz létre.
- Legyen egy osztályszintű `getCheapestFruit` metódus, ami visszaadja az előbbi `Fruit` objektumot.

A feladat részletes ismertetése (piac)

- Készítsd el a `market.Market` osztályt, ami egy egyszerűsített piacot reprezentál.
- Az osztály egy rejtett láncolt lista típusú adattagban tartsa nyilván, hogy milyen gyümölcsöket lehet kapni a piacon (`Fruit` típusú objektumok).

A feladat részletes ismertetése (piac)

- Az osztálynak egy publikus konstruktora legyen, ami egy fájlnevet kap paraméterként, amely gyümölcsöket tartalmaz. A metódus dolgozza fel a fájlt, szűrje ki belőle a hibás adatokat, majd töltsse fel a gyümölcsöket a láncolt listába. Ha a fájl nem létezik, vagy nem olvasható, akkor a gyümölcsök listája legyen üres, a konstruktor pedig ne engedje ki a keletkező kivételt.
 - Az inputfájl minden sora egy gyümölcsöt tartalmaz `név, ár` formában.
 - Ha a sor nem ilyen szerkezetű, vagy az `ár` nem konvertálható számmá, vagy a megadott adatokból a `Fruit` objektum nem hozható létre, akkor a sort figyelmen kívül kell hagyni és a feldolgozást a következő sorral kell folytatni.

A feladat részletes ismertetése (piac)

- Legyen egy `numberOfFruits` metódus, amely visszaadja a piacon még kapható gyümölcsök számát. Kezdetben az összes gyümölcs kapható.
- Legyen egy paraméter nélküli `show` metódus, ami szöveges típusban visszaadja a még kapható gyümölcsöket. A szöveg összeállításakor a gyümölcsök olyan sorrendben szerepeljenek, amilyen sorrendben a konstruktor beolvasta őket, és olyan formában, ahogy a `Fruit show` metódusa előállítja, a gyümölcsök között pedig legyen egy sortörés. Figyeljünk rá, hogy az utolsó gyümölcs után már ne legyen sortörés!

A feladat részletes ismertetése (piac)

- Legyen egy `cheaperThan` publikus metódus, ami egy `Fruit` típusú objektumot vár, és ilyen objektumok láncolt listáját adja vissza. A metódus gyűjtse össze azokat a gyümölcsöket, amelyek olcsóbbak, mint a paraméterben kapott gyümölcs. A gyümölcsök a visszaadott listában olyan sorrendben szerepeljenek, ahogy a konstruktor beolvasta őket.
- Legyen egy `average` publikus metódus, ami egy valós számot (`double`) adjon vissza, azt, hogy mennyi a gyümölcsök átlagos ára. Ha nincsenek gyümölcsök, akkor a metódus adjon vissza `-1`-et.
Figyelem: ha két egész számot osztunk egymással, akkor az eredmény mindig egész lesz (lefelé kerekít).

A feladat részletes ismertetése (piac)

- Legyen egy `buyCheapestFruit` publikus metódus, aminek nincs paramétere, és egy vásárlást szimulál. Egy vásárló mindig a legolcsóbb terméket veszi meg. Ha a legolcsóbb gyümölcssel azonos árú is van, akkor azt, amelyiket a konstruktor korábban vett fel a listába. A megvásárolt gyümölcsöt el kell távolítani a listából, és ez a gyümölcs lesz a metódus visszatérési értéke is. Ha a vásárlás elején egyetlen gyümölcs sem volt már a piacon, akkor a metódus `null`-t adjon vissza.

A feladat részletes ismertetése (piac)

- Legyen egy `sale` publikus metódus, aminek nincs paramétere és egy kiárusítást szimulál.
A kiárusítás addig tart, amíg el nem fogy az összes gyümölcs. A metódus egy gyümölcsöket tartalmazó láncolt listába gyűjtse össze, hogy a gyümölcsöket milyen sorrendben vásárolták meg. Ez a lista lesz a metódus visszatérési értéke.

A feladat részletes ismertetése (főprogram)

- Készítsünk egy `Main` osztályt, amelyben legyen egy főprogram.
- A főprogram egy paramétert vár: az inputfájl nevét. Ennek a segítségével hozzon létre egy `Market` objektumot.
- Ha a program nem kapott paramétert, akkor írjon ki hibaüzenetet.
- Hívja meg a `Market` objektum `sale` metódusát és a visszakapott lista minden egyes elemét írja ki a képernyőre (a `show` metódus segítségével).

Tesztelés

- Ha a program paraméterének a **LINK**-en található minta-inputot adjuk meg, akkor a következő kimenetet kell kapnunk:

```
barack (55 Ft)
korte (130 Ft)
szilva (130 Ft)
dinnye (2 015 Ft)
```