



Szoftverfejlesztés
gyakorlat – 13. óra

November 25

2019

Összefoglaló projektfeladat – Ingatlan és BigData

Feladatlap

1. Feladat – Ingatlan

Írj egy **Ingatlan** interfészt, ami egy paraméter nélküli `osszesKoltseg` metódust tartalmaz. A metódus az ingatlan költségét adja majd vissza.

Írj egy **Berelheto** interfészt, aminek három metódusa van. Az egyik a `mennyibeKerul`, ami egy egész számot (hónapok száma) kap paraméterül, és egy számmal (a bérleti díjjal) tér majd vissza. A másik a paraméter nélküli `lefoglaltE`, ami egy logikai értékkel tér majd vissza, hogy lefoglalták -e már az adott dolgot. A harmadik `lefoglal`, ami egy számot (lefoglalni kívánt hónapok száma) vár paraméterül, és logikai értékkel tér vissza attól függően, hogy sikeres volt -e a foglalás.

Írj egy **Lakas** absztrakt osztályt, ami implementálja az **Ingatlan** interfészt.

- Az osztály a következő adattagokkal rendelkezik: `terulet` (terület), `szobakSzama` (szobaSzam), `lakokSzama` (lakok), `negyzetmeterAr` (nmAr). Az adattagok legyenek láthatóak a leszármazott osztályokban is! Az osztály rendelkezzen paraméteres konstruktorral, ami beállítja az adattagok értékét.
- Készíts egy egész paramétert váró `bekoltozik` absztrakt metódust, ami embereket költöztet a lakásba. A metódus logikai értékkel tér vissza attól függően, hogy sikeres volt -e a beköltözés.
- Valósítsd meg az interfész metódusát, ami a terület és a négyzetméterenkénti ár alapján visszaadja a lakás teljes költségét.
- Készíts egy paraméter nélküli `lakokSzama` metódust, ami visszatér a lakásban lakók számát.
- Készíts egy `toString` metódust, ami Stringgé alakítja az osztály tulajdonságait.

Írj egy **Alberlet** osztályt, ami a **Lakas**-ból származik és implementálja a **Berelheto** interfészt.

- Az örökölteken kívül egy `foglaltHonapokSzama` (honapSzam) adattaggal rendelkezik, ami megadja, hogy hány hónapra van lefoglalva az albérlet. Készíts paraméteres konstruktort, ami az `Ősosztály` hasonló konstruktorát használva beállítja az adattagok értékét. A `foglaltHonapok` és a `lakokSzama` kezdeti értéke 0 legyen!
- Implementáld az interfész első metódusát. Az albérlet egy főre jutó havi költsége a lakás összes költségének és a lakók számának hányadosa, a metódus ez alapján adja öröklődés 125 vissza a kívánt időszak költségét. Ha nincs a lakásban lakó, úgy -1 legyen a visszatérési érték!
- Implementáld az interfész második metódusát. Az albérlet akkor nincs lefoglalva, ha a `foglaltHonapokSzama` 0.
- Implementáld az interfész harmadik metódusát. A lakás foglaltságától függően, módosítsd a `foglaltHonapokSzama` (ha még nem volt lefoglalva), és térj vissza megfelelő értékkel.
- Implementáld az `Ősosztály` absztrakt metódusát az alábbiak szerint: az albérletben egy szobában maximum 8 fő lakhat, és egy főre minimum 2 m² területnek kell jutnia. A beköltözés, és a metódus visszatérése ettől a feltételtől függjön. Ha a feltétel teljesül, úgy módosítsd a `lakokSzama` is.
- Írd felül a `toString` metódust, hogy már lefoglalt hónapok száma is szerepeljen benne. A metóduson belül használd az `Ősosztály` `toString`-jét is.

Írj egy **CsaladiApartman** osztályt, ami a **Lakas**-ból származik.

- Az örökölteken kívül egy `gyerekekSzama` adattaggal rendelkezik, ami megadja, hogy az összes lakóból mennyi a gyerek. Készíts paraméteres konstruktort, ami az `Ősosztály` hasonló konstruktorát használva beállítja az adattagok értékét. A `lakokSzama` és `gyerekekSzama` kezdeti értéke is 0 legyen.

- Készíts egy paraméter nélküli gyerekSzuletik metódust. A metódus ellenőrizze le, hogy van -e két felnőtt lakója az apartmannak, és ha igen, úgy növelje a lakók és gyerekek számát is 1-el. Térjen vissza logikai értékkel attól függően, hogy megszületett –e a gyerek.
- Implementáld az Őosztály absztrakt metódusát az alábbiak szerint. az apartman egy szobájában maximum 2 fő lakhat, és egy főre minimum 10 m² területnek kell jutnia. Gyerekek ebből a szempontból fél főnek számítnak csak. A beköltöző lakók mind felnőttek. A metódus visszatérése a fenti feltételtől függjön. Ha a feltétel teljesül, úgy módosítsd a lakók számát is.
- Írd felül a toString metódust, hogy a gyereke száma is szerepeljen benne. A metóduson belül használd az Őosztály toString-jét is.

Írj egy **Garazs** osztályt, ami implementálja mindkét fenti interfészt.

- Az osztály a következő adattagokkal rendelkezik: terület (terulet), négyzetméter ár (nmAr), fűtött -e (futottE), foglalt hónapok száma (honapSzam), áll -e benne autó (auto). Készíts paraméteres konstruktort, ami beállítja az adattagok értékét. A garázsban kezdetben ne álljon autó, és a foglalt hónapok száma 0 legyen!
- Implementáld az Ingatlan interfész metódusát! A garázs összes költségét a terület és a négyzetméterenkénti ár alapján kapot.
- Implementáld a Berelheto interfész metódusait! A garázs havi költsége a terület és ár alapján számolható, ezt még 1,5-el kell szorozni, ha fűtött is. Ez alapján visszadható a kívánt időszak költsége. A garázs akkor foglalt, ha a foglalt hónapok száma nagyobb, mint 0, vagy áll benne autó. A garázs foglaltságától függően módosítsd a foglalt hónapok számát (ha még nem volt lefoglalva), és térj vissza megfelelő értékkel.
- Készíts egy paraméter nélküli autoKiBeAll metódust. Ha már áll autó a garázsban, úgy az álljon ki, ha pedig nem áll, úgy álljon be. A metódus ne térjen vissza semmivel.
- Készíts egy toString metódust, ami Stringgé alakítja az osztály tulajdonságait.

Írj egy **Tarsashaz** nevű osztályt.

- Az osztálynak egy adattagja legyen: egy kollekció (pl. lista), ami képes Lakas és Garazs objektumokat is tárolni, valamint két egész szám, hogy maximálisan hány lakás és garázs lehet a társasházban. Írj konstruktort, ami a lakások és garázsok maximális számát kapja paraméternek, és beállítja a megfelelő adattagoat. Kezdetben a nincs lakás és garázs a házban.
- Írj két metódust lakasHozzaad és garazsHozzaad néven. Mindegyik metódussal a neki megfelelő objektumot helyezhetjük majd el a házban, így paraméternek mindegyik a megfelelő típust várja. A metódusok adják hozzá a kollekcióhoz a paraméterként kapott objektumot, ha még nem értük el az adott típusból a maximális mennyiséget. Visszatérési értékük logikai legyen attól függően, hogy sikerült -e a hozzáadás, vagy sem.
- Írj egy paraméter nélküli osszesLako metódust, ami visszadja, hogy a házban található lakásokban összesen hány lakó van.
- Írj egy paraméter nélküli ingatlanErtek metódust. A metódus számolja végig a házban lévő és használatban lévő lakások és garázsok értékét, és ezek összegével térjen vissza. Lakások esetén azok vannak használatban, amikben legalább 1 lakó lakik, míg garázs esetén az, ami le van foglalva.

Írj egy **Hazmester** nevű futtatható osztályt.

- Az osztály rendelkezzen egy karbantart statikus függvénnyel, ami egy fájl elérési útját várja paraméterül, visszatérési értéke pedig void. A metódus példányosítson egy Tarsashaz-at. A metódus feladata ezután,

hogy a fájlból beolvasott sorokat (ingatlanokat) feldolgozza, és létrehozson belőlük Alberlet, CsaladiApartman és Garazs objektumokat. Ezeket az objektumokat a megfelelő függvény használatával adja is hozzá a létrehozott társasházhoz. Miután végzett a fájl feldolgozásával, írja ki konzolra, hogy mennyi a társasház összes értéke. Minden esetleges kivételt (főleg FileNotFoundException és IOException) kezelj le kivételspecifikációval, vagy try blokkba.

Egy minta fájl felépítése az alábbi:

```
Alberlet 50.2 5 13000
CsaladiApartman 62.8 2 40000
Garazs 10.3 5000 futott
```

2. Feladat – Jelentkezők szűrése

Adott egy cég alkalmazottainak (employees) és a jelentkezőinek (applicants) egy listája. A listát a HR osztályon dolgozó munkatársak állították össze, úgy hogy végignézték az összes életrajzt és motivációs levelet. A cég szempontjából legfontosabb információkat kiszűrték és egységes formára hozták.

Feladata, hogy értékelje a jelentkezőket, majd keresse meg a 10 legjobbat közülük. A projektnek HRTamogato nevet adja meg.

A következő információkat tarjuk nyilván:

Alkalmazottak (300 fő)

- Keresztnév
- Vezetéknév
- Neme
- Hobbi
- Képességek
- Előléptetés

Jelentkezők (5000 fő)

- Keresztnév
- Vezetéknév
- Neme
- Hobbi
- Képességek

A feladat megoldásához az egyes alkalmazottakat és jelentkezőket értékelni, azaz valahogyan pontozni kell.

- Ehhez azt az információt adta a HR, hogy számukra csak a fontos tulajdonságokat rögzítették. Így egy tulajdonság értéke a személyek előfordulásában lévő darabszámmal, azaz a tulajdonság gyakoriságával határozható meg.
- A képességek és a hobbik előfordulására kell koncentrálni, így elegendő azokat összeszámolni súlyozottan az alkalmazottak listájából.

A projektet jól áttekinthetően tagold csomagokba:

- egysegek – entity
 - Ember – Absztrakt őssztály
 - Alkalmazott

- Jelentkezo
- feldolgozo – process
 - Feldolgozo
 - AlkalmazottFeldolgozo
 - JelentkezoFeldolgozo