

1. Írjon egy **Ingatlan** interfészt, ami egy paraméter nélküli *osszesKoltseg* metódust tartalmaz. A metódus az ingatlan költségét adja majd vissza.
2. Írjon egy **Berelheto** interfészt, aminek három metódusa van. Az egyik a *mennyibeKerul*, ami egy 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.
3. Írjon egy **Lakas** absztrakt osztályt, ami implementálja az **Ingatlan** interfészt.
  - Az osztály a következő adattagokkal rendelkezik: terület (*terulet*), szobák száma (*szobaSzam*), lakók száma (*lakok*), négyzetméter ár (*nmAr*). Az adattagok legyenek láthatóak a leszármazott osztályokban is. Az osztály rendelkezzen paraméteres konstruktorral, ami beállítja az adattagok értékét.
  - Készítsen 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.
  - Valósítsa meg az interfész metódusát, ami a terület és a négyzetméterenkénti ár alapján visszaadja a lakás teljes költségét.
  - Készítsen egy paraméter nélküli *lakokSzama* metódust, ami visszaadja a lakásban lakók számát.
  - Készítsen egy *toString* metódust, ami stringgé alakítja az osztály tulajdonságait.
4. Írjon egy **Alberlet** osztályt, ami a **Lakas**-ból származik és implementálja a **Berelheto** interfészt.
  - Az örökölteken kívül egy foglalt hónapok száma (*honapSzam*) adattaggal rendelkezik, ami megadja, hogy hány hónapra van lefoglalva az albérlet. Készítsen paraméteres konstruktort, ami az *ősosztály* hasonló konstruktorát használva beállítja az adattagok értékét. A foglalt hónapok és a lakók száma kezdeti értéke 0 legyen.
  - Implementálja 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 vissza a kívánt időszak költségét. Ha nincs a lakásban lakó, úgy 1 legyen a visszatérési érték.
  - Implementálja az interfész második metódusát. Az albérlet akkor nincs lefoglalva, ha a foglalt hónapok száma 0.
  - Implementálja az interfész harmadik metódusát. A lakás foglaltságától függően, módosítsa a foglalt hónapok számát – ha még nem volt lefoglalva –, és térjen vissza megfelelő értékkel.
  - Implementálja az *ősosztály* absztrakt metódusát az alábbiak szerint: az albérletben egy szobában maximum 8 fő lakhat, és egy főre minimum 2 m<sup>2</sup> területnek kell jutnia. A beköltözés és a metódus visszatérése ettől a feltételtől függjön. Ha a feltétel teljesül, úgy módosítsa a lakók számát is.
  - Írja felül a *toString* metódust, hogy a már lefoglalt hónapok száma is szerepeljen benne. A metóduson belül használja az *ősosztály toString*-jét is.
5. Írjon egy **CsaladiApartman** osztályt, ami a **Lakas**-ból származik.
  - Az örökölteken kívül egy *gyerekekSzama* adattaggal rendelkezik, ami megadja, hogy az összes lakóból mennyi a gyerek. Készítsen paraméteres konstruktort, ami az *ősosztály* hasonló konstruktorát használva beállítja az adattagok értékét. A lakók és gyerekek száma kezdeti értéke is 0 legyen.
  - Készítsen 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-gyel. Térjen vissza logikai értékkel attól függően, hogy megszületett-e a gyerek.
  - Implementálja az *ősosztály* absztrakt metódusát az alábbiak szerint. Az apartman egy szobájában maximum 2 fő lakhat, és egy főre minimum 10 m<sup>2</sup> területnek kell jutnia. Gyerekek ebből a szempontból fél főnek számítanak csak, egy gyereknek 5 m<sup>2</sup> terület is elegendő. A beköltöző lakók mind felnőttek. A metódus visszatérése a fenti feltételtől függjön. Ha a feltétel teljesül, úgy módosítsa a lakók számát is.

- Írja felül a *toString* metódust, hogy a gyerekek száma is szerepeljen benne. A metóduson belül használja az *ösosztály toString*-jét is.
6. Írjon 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 (*hnapSzam*), 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!
  - Implementálja az **Ingatlan** interfész metódusát. A garázs összes költségét a terület és a négyzetméterenkénti ár alapján kapja.
  - Implementálja a **Berelhető** interfész metódusait. A garázs havi költsége a terület és ár alapján számolható, ezt még 1,5-del kell szorozni, ha fűtött is. Ez alapján visszaadható a kívánt időszak költsége. A garázs akkor foglalt, ha a foglalt hónapok száma nagyobb, mint 0, vagy áll benne autó. A garázs foglaltságától függően módosítsa a foglalt hónapok számára – ha még nem volt lefoglalva –, és térjen vissza megfelelő értékkel!
  - Készítsen 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.
  - Készítsen egy *toString* metódust, ami stringgé alakítja az osztály tulajdonságait.
7. Írjon egy **Társasház** nevű osztályt.
- Az osztálynak egy adattagja legyen: egy kollekció, ami képes **Lakas** és **Garazs** objektumokat is tárolni, valamint két egész szám, hogy maximálisan hány lakás és garázs lehet a társasházban. Írjon konstruktort, ami a lakások és garázsok maximális számát kapja paraméternek, és beállítja a megfelelő adattagokat. Kezdetben nincs lakás és garázs a házban.
  - Írjon 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.
  - Írjon egy paraméter nélküli *osszesLako* metódust, ami visszaadja, hogy a házban található lakásokban összesen hány lakó van.
  - Írjon 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.
8. Írjon 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 példányosítsa egy **Társasház**-at. A metódus feladata ezután, hogy a fájlból beolvasott sorokat (ingatlanokat) feldolgozza, és létrehozzon 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. Ezt követően minden létrehozott metódust teszteljen.

Egy minta fájl felépítése az alábbi:

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