Konzol alkalmazás

**A) Konzol:** Vásárló **osztály létrehozása**

Hozzon létre egy osztályt a vásárlók tárolására. Minden vásárlónak ismert a neve, email címe és keretösszege pénznem értékben, amit a vásárlásokra szánhat.

* A név módosítható legyen, az email cím és a keretösszeg viszont nem változtatható meg a vásárló létrehozása után közvetlenül.
* A vásárló létrehozásakor az email címet és a keretösszeget kötelező megadni.
* A vásárló nem jöhet létre negatív keretösszeggel. Ilyenkor a program dobjon kivételt (InvalidOperationException vagy saját NegativeBalanceException).
* A nulla keretösszeg megengedett, de ebben az esetben a vásárló nem tud vásárolni.

**Feladatok**

**1. feladat**

* Hozzon létre egy vásárlót -500 Ft keretösszeggel. Jelenítse meg a képernyőn a keletkezett kivétel szövegét!
* Hozzon létre egy érvényes vásárlót, és írja ki az adatait! A vásárló adatai a következő formátumban jelenjenek meg:

Vásárló Valér ([vasarlo.valer@sokatveszek.hu](mailto:vasarlo.valer@sokatveszek.hu)) -> 20000 Ft

**2. feladat**

A keretösszeg pozitív értékkel növelhető, és az osztály tartsa nyilván, hogy hányszor történt növelés. Nem megfelelő összeggel történt növelés esetén dobjon kivételt!

**Feladat:**

* Növelje Valér keretösszegét 5000 Ft-tal, majd 10 000 Ft-tal.
* Írja ki a képernyőre Valér objektumát használva a keretösszeg-növelések számát!

A vásárlóról megállapítható legyen, hogy:

* Vásárolhat-e (van-e legalább 1 Ft keretösszege).
* Egy adott összegnél többet vásárolhat-e (pl. egy adott termék árát megengedheti-e magának).

**Feladat:**

* Írja ki a képernyőre, hogy Valér meg tudja-e vásárolni egy 45 000 Ft-os monitort!

**3. feladat**

Fejlessze ki a vásárlás lehetőségét a vásárló osztályban!

* Csak pozitív pénzösszegű vásárlás engedélyezett.
* A vásárló nem vásárolhat többet, mint amennyi pénze van, ellenkező esetben dobjon kivételt.

**Feladatok:**

* Vásároljon Valér -1000 Ft értékben, és kapja el a keletkezett kivételt
* Vásároljon Valér 200 000 Ft értékben, és kapja el a keletkezett kivételt!
* Vásároljon Valér 10 000 Ft értékben, majd írja ki a képernyőre Valér új keretösszegét!

**4. feladat**

Hozzon létre még négy vásárlót, amelyek közül egynek nulla a keretösszege!   
Írja ki az adataikat a képernyőre a megadott formátumban.

## ****B)**** Konzol: több adat tárolására alkalmas osztály

**1. feladat - Vásárlók tárolása és kezelése**

Készítsen egy osztályt, amely több vásárló tárolására alkalmas.

* A vásárlók adatai közvetlenül ne legyenek írhatók, csak lekérdezhetők.
* Az osztályban legyen lehetőség új vásárló hozzáadására.
  + Ne lehessen két vásárlónak ugyanaz az email címe.
  + Ha mégis megpróbálunk duplikált email címmel vásárlót hozzáadni, a rendszer dobjon kivételt!
* Biztosítsa a vásárlók törlését email cím alapján lehetőséget.
  + Ha az adott email cím nem létezik, a rendszer dobjon kivételt!

**2. feladat - Vásárlók hozzáadása és törlése**

* Vegye fel Valér és a további négy vásárló adatait a vásárlókat tároló objektumba!
* Írja ki a képernyőre a tárolt vásárlók számát és adataikat!
* Töröljön egy vásárlót és írja ki újra a tárolt vásárlók számát!

**3. feladat - Vásárlók adatainak lekérdezése**

Készítsen metódusokat az osztályhoz a következő információk lekérdezésére, és írassa ki az eredményeket a főprogramban:

1. A legnagyobb keretösszeggel rendelkező vásárló(k) neve és keretösszege. Ha több vásárlónak azonos a legnagyobb keretösszege, mindet írja ki.
2. A legkisebb keretösszeggel rendelkező vásárló(k) email címe(i). Ha több vásárlónak azonos a legkisebb keretösszege, mindet írja ki.
3. A pozitív keretösszeggel rendelkező vásárlók keretösszegeinek átlaga, ezres kerekítéssel.
   * A 0 Ft-os vásárlók kihagyása az átlagolásból.
   * Az átlag értéket matematikailag kerekítse ezresekre!
4. Generáljon a főprogramban két számot a legnagyobb és legkisebb keretösszeg között!
   * Írja ki azokat a vásárlókat, akik ezen két szám között tudnak vásárolni keretösszegük alapján!
   * Ha nincs ilyen vásárló, azt is jelezze a program.
   * Írja ki a rendezett listát a vásárlók neveivel és keretösszegeikkel.
5. Összes vásárló teljes keretösszegének kiszámítása és az eredmény kiírása.
6. Keresés email cím alapján.
   * Írjon egy metódust, amely egy adott email cím alapján visszaadja a vásárló adatait.
   * Ha a vásárló nem található, a program jelezze ezt megfelelően.
7. A vásárlók százalékos eloszlása keretösszeg kategóriák szerint.
8. Kategóriák:
   * 0 Ft\*\* (pl. nincs pénze)\*\*
   * 1 - 10 000 Ft
   * 10 001 - 50 000 Ft
   * 50 001 Ft felett Számítsa ki, hogy a vásárlók hány százaléka tartozik az egyes kategóriákba, és írja ki a képernyőre.

## Adatbázis műveletek a repository rétegben

### I. Adatok lekérdezése

1. **Vásárlók neveinek lekérése:**  
   Készíts egy olyan metódust, amely az összes tárolt vásárló nevéből egy listát készít. (Pl.: List<string> GetCustomerNames())
2. **Email címek kinyerése:**  
   Írj egy metódust, amely az összes vásárló email címét adja vissza egy listában.
3. **Adatátalakítás egyedi típusba:**  
   Készíts egy metódust, amely a vásárlók bizonyos jellemzőit (például név, email és egyenleg) egy új típusú objektumba (például egy adatátviteli objektumba – DTO) alakítja, majd ezeket adja vissza.
4. **Formázott információ lekérése:**  
   Írj egy metódust, amely az összes vásárlóról egy formázott szöveget készít (pl. "Név: [név], Egyenleg: [egyensúly] Ft"), és ezt adja vissza egy listában.

### II. Rekordok szűrése

1. **Egyenleg alapján szűrés:**  
   Készíts egy metódust, amely visszaadja azokat a vásárlókat, akiknek az egyenlege meghalad egy adott összeget.
2. **Nulla egyenlegű vásárlók:**  
   Írj egy metódust, amely kiválasztja azokat a vásárlókat, akiknek az egyenlege pontosan 0 Ft.
3. **Név részleges egyezés:**  
   Készíts egy metódust, amely visszaadja azokat a vásárlókat, akik nevében szerepel egy adott részszöveg (keresés legyen kis- és nagybetű érzéketlen).
4. **Összetett feltételek alkalmazása:**  
   Írj egy metódust, amely két vagy több feltétel alapján szűri a vásárlókat. Például olyan vásárlók kiválasztása, akiknek az egyenlege nagyobb egy megadott értéknél ÉS az email címük egy adott domaint tartalmaz.

### III. Adatok rendezése

1. **Alfabetikus sorrend:**  
   Készíts egy metódust, amely visszaadja a vásárlókat a nevük alapján növekvő (ABC) sorrendben.
2. **Egyenleg szerinti rendezés:**  
   Írj egy metódust, amely a vásárlókat csökkenő sorrendben rendezi az egyenlegük alapján.
3. **Többszörös rendezés:**  
   Készíts egy metódust, amely először az egyenleg (csökkenő sorrend), majd a név (növekvő sorrend) szerint rendezi a vásárlókat.
4. **Email domain szerinti rendezés:**  
   Írj egy metódust, amely a vásárlók email címének domén részét használva rendezi őket. (Ehhez először szükséges kinyerni a domént az email címből.)

### IV. Ismétlődések megszüntetése

1. **Email domainek egyedisége:**  
   Készíts egy metódust, amely visszaadja az összes vásárló által használt email domaineket, ismétlődések nélkül.
2. **Egyedi egyenlegértékek:**  
   Írj egy metódust, amely a vásárlók között előforduló egyedi egyenlegértékeket adja vissza.
3. **Név kezdőbetűk:**  
   Készíts egy metódust, amely visszaadja a vásárlók neveinek első betűit úgy, hogy minden betű csak egyszer szerepeljen a listában.
4. **Egyedi rekordok összetett feltétellel:**  
   Készíts egy metódust, amely a vásárlókat olyan kombinált jellemzők alapján szűri (például név és egyenleg), hogy ha két vásárlónak ugyanaz a név és azonos egyenlege van, akkor csak egyszer szerepeljenek az eredményben. Ehhez használhattok egyéni összehasonlítást.

### V. Adatok csoportosítása

1. **Név első betűje alapján csoportosítás:**  
   Készíts egy metódust, amely a vásárlókat a nevük első betűje szerint csoportosítja. Az eredményben minden csoporthoz tartozzon a betű és a csoportba tartozó vásárlók listája.
2. **Egyenleg intervallumok szerinti csoportosítás:**  
   Írj egy metódust, amely a vásárlókat olyan kategóriákba sorolja, mint például "0 Ft", "1–20 000 Ft", "20 001 Ft felett". A csoportosítás segítségével a kategória nevét és az adott kategóriába tartozó vásárlók listáját adja vissza.
3. **Email domain alapján csoportosítás:**  
   Készíts egy metódust, amely az email cím doménje szerint csoportosítja a vásárlókat, így megkapva minden doménhez tartozó vásárlók listáját.
4. **Pozitív vs. Nulla egyenleg szerinti csoportosítás:**  
   Írj egy metódust, amely két csoportba osztja a vásárlókat: az egyik csoportba azok kerülnek, akiknek az egyenlege pozitív, a másikba pedig azok, akiknek 0 Ft az egyenlege.