

## Programozás 3. gyakorló feladat az első ZH-hoz

Készítsen egy egyrétegű konzolos alkalmazást, melynek segítségével egy lokális adatbázisban tárolt koncerthelyszín-adatokkal, és a program által véletlenszerűen generált jegyeladási adatokkal kell majd dolgoznia. A jegyeladási adatokból statisztikákat kell generálni, majd ez alapján el kell döntenie, hogy van –e hiba a jegyeladási adatokban (nem adtunk –e el több jegyet, mint amennyi hely van).

### ADATBÁZIS ÉS AZ ENTITÁSOK KIALAKÍTÁSA

Code First megközelítéssel egy külön Class Library projektben hozzon létre egy lokális adatbázis három táblával: `venues`, `sectors`, `sellers`.

a) A három entitás:

- a. Venue (helyszín)
  - i. `int Id` – elsődleges kulcs, adatbázis által szolgáltatott növekményes érték
  - ii. `string Name` – max 100 hosszú, megadása kötelező
- b. Sector (helyszínen belüli szektor)
  - i. `int Id` – elsődleges kulcs, adatbázis által szolgáltatott növekményes érték
  - ii. `int Capacity`
  - iii. `int VenueId` – idegen kulcs a Venues táblára tekintve
  - iv. `string Code` – max 13 hosszú, megadása kötelező
- c. Seller (a helyszínhez engedélyezett jegyeladó cég)
  - i. `int Id` – elsődleges kulcs, adatbázis által szolgáltatott növekményes érték
  - ii. `string Name` – max 100 hosszú, megadása kötelező
  - iii. `int VenueId` – idegen kulcs a Venues táblára tekintve

b) Attribútumok segítségével definiálja a szükséges táblastruktúrát, a fluent szintaxist használja az esetleges külső kulcsok / navigation property-k definiálása során (lehetséges attribútumok helyett teljesen a fluent api használata is, de ez hosszabb és több gépelést igényel).

c) Hozzon létre néhány mintaadatot az adatbázisban seed adatként (*Helyszín: Pap László Sportaréna; Szektorok: A, B, C, 1100, 2500, 1500 kapacitásként; Eladók: Broadway, Eventim*).

### ELADÁSI TRANZAKCIÓK ELŐÁLLÍTÁSA

Legyen egy olyan Class Library az alkalmazásban, amely képes véletlenszerű jegyeladásokat generálni napi felbontásban.

a) Egy nap adatát a `DailySale` osztály tárolja

- a. `DateTime Date`
- b. `string SellerName`
- c. `string SectorCode`
- d. `int TicketsSold`

b) Készítsen egy generáló osztályt `DailyStatGenerator` néven, amely

- a. konstruktorban megkapja az eladókat és a szektorokat, külön-külön `string` tömbként;
- b. legyen egy `GenerateList(int numDays, int numInstances, int maxSold)` metódusa, amely egy listányi `DailySale` objektummal tér vissza. A `numInstances` paraméter mondja meg, hogy hány `DailySale` példányt kell véletlenszerűen legenerálni; a `numDays` meghatározza, hogy a mai naptól visszafelé legfeljebb meddig mehet vissza véletlenszerűen a dátum; a `maxSold` pedig a példányokban lehetséges maximum véletlenszerű `TicketsSold` értéket jelzi. Ezek segítségével (valamint a konstruktorban kapott előadók és szektorok közül minden példányhoz egyet-egyet véletlenszerűen kiválasztva) generálja le a listát, tele véletlen jegyeladási példányokkal.

## XML GENERÁLÁSA

Hozzon létre egy olyan Class Library-t, amely képes XML generálására egy listányi `DailySale` példányból.

- Az XML gyökér eleme `stats` névre hallgasson;
- Napi bontásban (`day`) szektoronként (`sector`) és eladónként (`seller`) is egyaránt legyenek külön-külön összesítve a listából a jegyeladások (`sold`). Az adott nap értékét XML attribútumként jelenítsük meg a `ToShortDateString()` segítségével. A szektor azonosítását a kódjával, az eladót a nevének XML attribútumként való tárolásával oldja meg;
- Példa XML:

```
<stats>
  <day date="2020. 10. 01.">
    <sector code="B">
      <sold>149</sold>
    </sector>
    <sector code="A">
      <sold>58</sold>
    </sector>
    <sector code="C">
      <sold>199</sold>
    </sector>
    <seller name="Eventim">
      <sold>222</sold>
    </seller>
    <seller name="Broadway">
      <sold>184</sold>
    </seller>
  </day>
  <day date="2020. 10. 02.">
    <sector code="B">
      <sold>59</sold>
    </sector>
    <sector code="A">
      <sold>136</sold>
    </sector>
    <sector code="C">
      <sold>93</sold>
    </sector>
    <seller name="Eventim">
      <sold>141</sold>
    </seller>
    <seller name="Broadway">
      <sold>147</sold>
    </seller>
  </day>
  ...
</stats>
```

## PROGRAM

- Jelenítse meg az adatbázisból az összes szektort és az összes eladót;
- `DailyStatGenerator` segítségével készítsen egy listát a jegyeladásokról;
- az XML generátor segítségével állítson elő napi statisztikát XML formátumban;
- ezt követően az XML-t felhasználva:
  - Határozza meg, hogy eladónként összesen mennyi jegyet adtak el;
  - Határozza meg, hogy szektoronként összesen mennyi jegyet adtak e;
  - Határozza meg, hogy szektoronként mennyi jegy maradt;
  - Határozza meg, hogy van-e hiba az eladási listában (ha több jegyet adtak el bármelyik szektorban, mint a szektor kapacitása).

Jó munkát kívánunk!