

Nyelvvizsga – 25 pont

Egy nyelvvizsga központnál csak középfokú oklevelet lehet szerezni, viszont azt többféle nyelven is.

Választható A típusú (írásbeli), B típusú (szóbeli) és C típusú (komplex) nyelvvizsga. Az írásbelin legfeljebb 90 pont, a szóbelin legfeljebb 30 pont, a komplex vizsgán pedig legfeljebb 120 pont szerezhető.

A **Nyelvvizsga** nevű C# projektből kiindulva oldd meg a következő feladatokat! Kizárólag a **Vizsga.cs** és a **Statisztika.cs** állományokat kell módosítani! A **Program.cs**-ben a megírandó osztályok használatára látható példa. Ezt javasolt kommentbe tenni, amíg a szükséges kódrészletek nincsenek elkészítve.

A továbbiakban minden kért adattag és metódus publikus legyen, ha a feladat másképp nem kéri! További segéd függvényeket szabad készíteni, de azok csakis az adott osztályon belül legyenek láthatóak!

1. A **Vizsga** osztályban 3 **privát** adattag legyen, melyek neve: nyelv, típus, pont. Típusaik rendre szöveg, karakter és egész szám. **1 pont**
2. Készíts a **Vizsga** osztályhoz egy 3 paraméteres konstruktort! **1 pont**
3. Hozz létre Nyelv és Pont néven két property-t, továbbá mindkettőhöz egy-egy getter metódust! **1 pont**
4. Vegyél fel egy további szöveges property-t Típus néven, amelyhez tartozzon getter és setter is! **4 pont**
 - a. A setter segítségével beállíthatjuk az osztály típus adattagját! Ügyelj rá, hogy csakis „A”, „B” és „C” értékeket fogadj el a setterben!
 - b. A getter megadja a vizsga típusát szövegesen, vagyis A típusú nyelvvizsga esetén „írásbeli”, B-nél „szóbeli”, C-nél pedig a „komplex” szöveg a visszatérési értéke!
5. Tartozzon az osztályhoz Eredmeny(tizedes) néven egy **privát** metódus, amely a vizsga százalékos eredményét adja vissza, a paraméterként megkapott mennyiségű tizedesjegyre kerekítve! A *paraméter egész típusú*. **3 pont**
6. Írd felül az alapértelmezett ToString() metódust, hogy szöveggé alakítva egy vizsga adatait a következőt kapjuk:

<nyelv> (<típus>): <eredmeny>%

ahol <adattag> jelöli a megjelenítendő adattagot, például: „angol (komplex): 54,2%”. Az eredményt 1 tizedesre kerekítsd! **2 pont**

7. A **Statisztika** osztályon belül vegyél fel egy vizsgák nevű **privát** listát, amely **Vizsga** típusú objektumokat tárol! **1 pont**
8. A **Statisztika** osztály konstruktora egyetlen szöveges paramétere legyen egy fájl neve! Az **adatok.txt** fájl minden sora egy vizsga eredményét tartalmazza, pontosvesszővel tagolva.

Az A és B típusú vizsgák esetén soronként 3 adat található: a nyelv, a típus és a pontszám. C típusú vizsgáknál viszont 4 adatot tartalmaz egy sor, külön szerepel az írásbeli és a szóbeli részre kapott pont, amelyek összegét kell tárolni, mint a vizsga végső pontszáma.

Minta (adatok.txt):

```
francia;C;43;21
német;A;50
olasz;C;77;11
angol;C;65;15
angol;C;56;21
...
```

Vagyis az 1. sor szerint francia nyelvből tettek egy komplex nyelvvizsgát, amely 64 pont a 120-ból. Ezután a 2. sor alapján németből egy írásbeli vizsgán 50 pontot értek el a 90-ből.

Olvasd be az adatokat, és tárold el a vizsgák listában!

4 pont

9. Adj a **Statisztika** osztályhoz egy **LegjobbKomplex()** nevű metódust, amely visszatérési értéke a komplex vizsgák közül a legjobban sikerült vizsga! *Feltehető, hogy van a beolvasott adatok között komplex vizsga.* **3 pont**
10. Készíts egy **Kiir(fajl)** nevű metódust, ami létrehoz egy szöveges fájlt a megadott névvel, és beleírja a vizsgák eredményeit nyelv szerint abc sorrendben! A rendezést a **buborékos rendezés** algoritmusával végezd el! **5 pont**

Minta (eredmenyek.txt):

```
angol (komplex): 66,7%
angol (komplex): 64,2%
angol (komplex): 88,3%
angol (írásbeli): 94,4%
angol (írásbeli): 61,1%
...
olasz (szóbeli): 83,3%
olasz (komplex): 90,8%
spanyol (írásbeli): 47,8%
spanyol (szóbeli): 60%
spanyol (komplex): 60%
spanyol (írásbeli): 51,1%
```

A szöveges értékek összehasonlításához használd a **String** osztály `CompareTo()` metódusát, amelynek hivatalos dokumentációja:

C# Copy

```
public int CompareTo(object? value);
```

Parameters

value `Object`

An object that evaluates to a `String`.

Returns

`Int32`

A 32-bit signed integer that indicates whether this instance precedes, follows, or appears in the same position in the sort order as the `value` parameter.

Expand table

Value	Condition
Less than zero	This instance precedes <code>value</code> .
Zero	This instance has the same position in the sort order as <code>value</code> .
Greater than zero	This instance follows <code>value</code> . -or- <code>value</code> is <code>null</code> .