



ESZTERHÁZY KÁROLY EGYETEM

Magasszintű programozási nyelvek II.

2020/2021 tavaszi félév
1. zárthelyi dolgozat

Általános információk

Készítsen **konzol alkalmazást, melynek neve az Ön Neptunkódja!** Ebben oldja meg a következő oldalakon kijelölt feladatokat! Miután a feladatokkal végzett, a programot **tömörítse be (.zip, .rar, .7z), és annak szintén legyen a neve az Ön Neptunkódja!** Az így elkészített tömörített állományt töltsse föl az e-learning portálon látható feltöltési lehetőségnél!

A feltöltés határideje:

2021. március 12. 20 : 00

A dolgozat írása közben az órai anyagok felhasználhatóak.

A dolgozat írása közben ugyanazon a platformon, ahol az órák szoktak lenni, Ön köteles bejelentkezve lenni. Amennyiben a gyakorlatvezető jelzi szándékát, Önnek a lehető leggyorsabban (másodpercekről beszélünk) meg kell osztania a képernyőjét és kameraképét, hogy leellenőrizzük, nem kap külső segítséget!

Törekedjen részmegoldásokra! Például ha nem tudja a cégnév esetén megvizsgálni, hogy legfeljebb egyszer módosították-e, akkor a property maradék részét oldja meg, és haladjon tovább!

Feladatkiírás

A megvalósítandó feladatban egy fiktív Volán társaság útvonaltervezésének az implementálása a feladat.

Autóbusz osztály

Hozzon létre egy osztályt **Autobusz** néven, majd implementálja az alábbiakat!

Két autóbuszt akkor tekintünk egyenlőnek, ha megegyezik a gyártója és a modellje.

Mezők és property-k (adatok kezelése)

- **Gyártó:** Kívülről nem módosítható szöveges típus. Nem lehet null, hossza legalább 2 karakter.
- **Modell:** Kívülről nem módosítható szöveges típus. Nem lehet null, hossza legalább 2 karakter.
- **Életkor:** kívülről nem módosítható egész típus. Nem lehet negatív.
- **Gyári fogyasztás (l/100km):** kívülről nem módosítható valós típus. Nem lehet negatív. Érvényes a [9,5; 16,8) intervallumban.

Metódusok és property-k

- **Tényleges fogyasztás:** Csak olvasható property, mely becslést ad a tényleges fogyasztásra. Ennek képlete: $gyarifogyasztas \cdot (1 + életkor/20)$
- **ToString:** Írja felül a ToString() metódust úgy, hogy az adott autóbusz minden adata ízlésesen jelenjen meg!

Konstruktorok

- **Konstruktor 1:** Kérje be a gyártó, modell, életkor és gyári fogyasztás adatokat paraméterben, majd ezeket mentse el!
- **Konstruktor 2:** Kérje be a gyártó, modell és gyári fogyasztás adatokat paraméterben! Hívja meg a *Konstruktor* 1-et a megfelelő értékekkel, mely során az életkort 0-ra állítsa be!

Volán osztály

Hozzon létre egy osztályt **Volan** néven, majd implementálja az alábbiakat!

Mezők és property-k (adatok kezelése)

- **Autóbuszok:** autóbuszok tárolására alkalmas lista, megfelelő védelemmel ellátva.
- **Cégnév:** kívülről nem módosítható szöveges mező. Nem lehet null, legfeljebb egyszer módosítható, legalább 5 karakter, tartalmaznia kell legalább egy *space* karaktert.

Metódusok és property-k

- **Autóbusz hozzáadása:** publikus metódus, melynek feladata, hogy elkészítsen egy autóbusz példányt, majd elhelyezze azt a listában. Paraméterben kérje be a gyártó,

modell, életkor és gyári fogyasztás autóbusz-adatokat! Amennyiben van már ilyen autóbusz a listában és fiatalabb is, úgy dobjon kivételt!

- **Indexer:** Írjon egy indexert, mely visszaadja az i . buszt a listából! Ha az index értéke nem megfelelő, dobjon kivételt!

Konstruktorok

- **Konstruktor 1:** Kívülről nem hívható paraméter nélküli konstruktor. Célja, hogy inicializálja az autóbuszok listáját.
- **Konstruktor 2:** Bekéri a cég nevét és eltárolja azt. Az autóbuszok inicializálását a *Konstruktor 1* meghívásával valósítja meg.

Útvonal osztály

Hozzon létre egy osztályt **Útvonal** néven, majd implementálja az alábbiakat!

Mezők és property-k (adatok kezelése)

- **Települések:** településnevek tárolására alkalmas lista, megfelelő védelemmel ellátva.
- **Távolságok:** valós értékű lista, mely az előző listában egymást követő települések távolságait (km) tárolja, megfelelő védelemmel ellátva.

Metódusok és property-k

- **Első település hozzáadása:** Publikus metódus, mely egy településnevet kap paraméterként. Feladata, hogy a települések listájához legelső elemként hozzáadja ezt a településnevet. Ha a települések listája eredetileg nem üres, úgy dobjon kivételt!
- **Település hozzáadása:** Publikus metódus, mely egy településnevet és egy távolságot kap paraméterként. Feladata, hogy a települések listájához és a távolságok listájához hozzáadja ezeket! Ha a települések listája eredetileg üres, úgy dobjon kivételt!
- **Fogyasztás:** publikus metódus, mely paraméterként megkap egy busz példányt és kiszámolja, hogy mennyi üzemanyagot használ el a busz az adott útvonalon! (A busz tényleges fogyasztásával számoljon!)

Főprogram

- Készítsen egy *Volan* példányt (pl. Nógrád Volán) és egy *Útvonal* példányt!
- A mellékelt csv fájlokat ("telepulesek.csv" és "buszok.csv") nyissa meg és olvassa be a bennük levő adatokat. A fájlok nevei és a fejlécek alapján egyértelmű az adatok tartalma és sorrendje.
- Írd ki a képernyőre annak a busznak az adatait, mellyel a legtakarékosabban tehető meg az út!