



ESZTERHÁZY KÁROLY KATOLIKUS EGYETEM

Magasszintű programozási nyelvek II.

2021/2022 tavaszi félév
1. zárthelyi dolgozat

Általános információk

Készítsen egy **mappát**, melynek neve az **Ön Neptunkódja!** Ebben oldja meg a kijelölt feladatokat, mely projektek nevei 01,02,03 legyenek! Miután a feladatokkal végzett, a teljes mappát **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 az e-learning felületen található.

A feltöltés helye (oldal alján):

<https://elearning.uni-eszterhazy.hu/course/view.php?id=4422>

A dolgozat írása közben az órai anyagok felhasználása nem megengedett!

A dolgozat írása közben nem használhat internetes segédanyagokat, és nem kommunikálhat senkivel!

A nem beazonosítható dolgozatok elégtelenek, igazolás hiányában nem újraírhatóak! Figyeljen arra, milyen néven adja be a mappát!

Bármilyen idegenkezűség, vagy nem megengedett segédanyag használata következtében a dolgozat érdemjegye automatikusan elégtelen.

Törekedjen rész megoldásokra! Ha valamit nem tud megoldani (pl egy property-t nem tud teljes mértékben kidolgozni), hagyja ott annak szignatúráját, vagy hozza olyan állapotra, hogy tudjon vele tovább dolgozni!

A dolgozat megírására 2 órája van, figyeljen az időbeosztásra!

Feladatkiírás

A zárthelyi dolgozatban egy fiktív használtautókereskedés által kínált járművek adatainak tárolását és azok lekérdezéseit menedzselő alkalmazást kell fejlesztenie.

Az osztályok amelyeket implementálnia kell majd az alábbiak szerint épülnek föl.

- Gépkocsi (őszosztály): a feladat nem foglalkozik vele, de valójában ez minden jellegű (személygépkocsi, tehergépkocsi, stb...őse)
 - Személygépkocsi (a gépkocsi gyermekosztálya)
- Autó kereskedés (konténerosztály a gépkocsik nyilvántartására)

Gépkocsi

Hozzon létre osztályt **Gepkocsi** néven, és implementálja benne az alábbiakat!

Mezők és property-k

- **Rendszám**: szöveges adat, az osztályon kívül nem módosítható. Értéke nem lehet null vagy üres string. Csak számokat, ékezetmentes nagybetűket és '-' karaktert tartalmazhat! Pontosan 7 karakter hosszú! Az első három karakter csak betű, míg az utolsó három karakter csak szám lehet. Az utolsó három karakter nem lehet csupa 0!
- **Évjárat**: egész típusú adat, kívülről nem módosítható. Értéke az [1950; 2022] intervallumban érvényes. Értéke csak egyszer adható meg. Ha tudja, implementálja úgy, hogy az intervallum felső határa a mindenkor aktuális évszám legyen! Ha ez nem megy, égesse be a 2022-öt!
- **Eredeti ár**: egész típusú adat, kívülről nem módosítható. Értéke a [300 000; 12 000 000] intervallumban érvényes. Értéke csak egyszer adható meg. **Ennél az adatnál írja meg a klasszikus property metódusokat a nyelvi szinten támogatott property-k helyett!**
- **Állapot**: felsorolás típus, értékei: **Ujszeru, Megkimelt, Serult, Hibas**
- **Kor**: egész értékű, csak olvasható property. Értéke az aktuális év és az évjárat különbsége!
- **Extra ár**: egész értékű, csak olvasható property. Amennyiben a gépkocsi legfeljebb 2 éves és újszerű, úgy az extra ár az eredeti ár 2%-a, egyébként 0 Ft. Alkalmazzon késői kötést!

Konstruktorok

- Készítsen konstruktort, mely az összes adatot (rendszám, évjárat, eredeti ár és állapot) bekéri és eltárolja.
- Készítsen konstruktort, mely minden adatot bekér a gépkocsi állapotán kívül! Ezt az előző konstruktor meghívásával automatikusan *megkímélte* állítja!

Metódusok

- Készítsen egész értékkel visszatérő késői kötésű metódust **VetelAr** néven a következők szerint. A vételárhoz először meg kell állapítani mennyit amortizálódik évente az autó az állapotától függően.

Állapot	Amortizáció
Ujszeru	9%
Megkimelt	10%
Serult	11%
Hibas	12%

A jármű értékét az

$$eredetiAr \cdot amortizacio^{gepjarmuKora} + extraAr$$

képlet segítségével tudja számolni!

- Írja felül a **ToString()** metódust úgy, hogy minden tárolt és számított adat ízlésesen megjelenjen!
- Írja felül az **Equals()** metódust! Két gépkocsi akkor egyezik meg, ha ugyanaz a rendszámuk.

Személygépkocsi

Hozzon létre osztályt **Szemelygepkocsi** néven, jelölje meg ősként a **Gepkocsi** osztályt és implementálja benne az alábbiakat!

Mezők és property-k

- **Szállítható személyek száma**: egész típusú adat, kívülről nem módosítható. Értéke a $\{2, 4, 5, 7\}$ halmaz valamely eleme.
- **Van vonóhorog**: logikai érték, nincs megkötés.
- **Klíma**: felsorolás típus, értékei: Nincs, Manualis, Digitalis, DigitalisTobbzonas
- Írja felül a **Gepkocsi** osztályban definiált **extra ár** property-t! A korábbi számítást egészítse ki azzal, a vonóhorog ára $60\,000\text{ Ft}$. Ha a szállítható személyek száma 7, az plusz $100\,000\text{ Ft}$, a klímák ára pedig az alábbi táblázat alapján számolható.

Klíma	Ár
Nincs	0 Ft
Manuális	$40\,000\text{ Ft}$
Digitális	$150\,000\text{ Ft}$
Digitális - több zónás	$350\,000\text{ Ft}$

Konstruktorok

- Készítsen konstruktort, mely az összes adatot (rendszer, évjárat, eredeti ár, állapot, szállítható személyek száma, vonóhorog és klíma) bekéri és eltárolja.
- Készítsen konstruktort, mely minden adatot bekér az gépkocsi állapotán és a klímán kívül! Ezt az előző konstruktor meghívásával automatikusan *megkímélte* és *digitális*-ra (nem többzónás) állítja!

Metódusok

- Írja felül a **Gepkocsi** osztályban definiált **VetelAr** metódust! Személygépkocsik esetén az amortizáció az alábbiak szerint alakul.

Állapot	Amortizáció
Ujszeru	8%
Megkimelt	9%
Serult	12%
Hibas	13%

Amennyiben a szállítható személyek száma 7, úgy az amortizáció mértékének számoljon az 1,2-szeresével! A korábban megadott képletet használja itt is a vételár kiszámításához!

- Írja felül a **ToString()** metódust úgy, hogy minden tárolt és számított adat ízlésesen megjelenjen!

Autó kereskedés

Hozzon létre osztályt **Kereskedes** néven, és implementálja benne az alábbiakat!

Mezők és property-k

- **Gépkocsik listája**: tetszőleges mennyiségű Gepkocsi típusú objektum tárolására alkalmas lista ellátva a megfelelő szintű védelemmel.
- **Személygépkocsik**: Összegyűjti és visszaadja a gépkocsik között megtalálható személygépkocsikat.
- **Legolcsóbb megkímélt személygépkocsi**: Megkeresi és visszatér a legolcsóbb olyan személygépkocsi objektummal, melynek állpota megkímélt! Vizsgálja meg, hogy van-e egyáltalán személygépkocsi a gépkocsik között!
- **Indexelő**: Készítsen *Gepkocsi* típusú objektummal visszatérő indexelőt, mely rendszám szám alapján adja vissza a keresett gépkocsit!

Metódusok

- Készítsen metódust **AddGepkocsi** néven, mely paraméterben kér egy *Gepkocsi* típusú objektumot! Ellenőrizze, hogy az adott gépkocsi szerepe-e már a rendszerben, mielőtt elmenti!
- Készítsen metódust **SzemelyGepkocsikAdottArig** néven, mely paraméterben bekér egy állapotot és egy maximum árat. Gyűjtse listába és térjen vissza azon személygépkocsik listájával, melyek megfelelnek a paraméterben kapott értékeknek!

Főprogram

- Hozzon létre egy **Kereskedes** példányt!
- Olvassa be a mellékelt **gepkocsik.csv** pontosvesszővel tagolt fájlt!
 - A fájl vegyesen tartalmaz gépkocsikat és személygépkocsikat. Ha az adott sor első adata "**G**", úgy a sor gépkocsit tartalmaz, míg "**SZ**" esetén személygépkocsit. Az adatok sorrendje megegyezik a dokumentumban definiált adatok sorrendjével!
 - Az egyes sorok esetén hozzon létre **Gepkocsi** vagy **Szemelygepkocsi** példányokat és azokat mentse el az *Kereskedes* **AddGepkocsi** metódusának hívásával!
- Jelenítse meg az összes a konténerosztályban implementált property és metódus eredményét a kijelzőn!

Szöveg konvertálása enummá:

```
MyEnum myEnum = (MyEnum)Enum.Parse(typeof(MyEnum), "text");
```