

Feladat – Kiszh 2 – FONTOS INFÓK

- A feladat során alkalmazd a megtanult objektum-orientáltsági elveket!
- A megadott példakódon ne módosíts, hacsak a feladat nem kéri! Ez alól kivétel a Program.cs elején lévő kommentek (#define). A megoldásnak ezen fájlokkal kell mennie, hiszen az ellenőrzés során a Moodle biztosítja őket.
- Figyelj a kiírás megfelelő formátumára, szóközökre, új sorokra! Az automata javító csak a tényleges kimenetet látja, nem tudja mit akartál.
- Minden pont értékeléséhez szükséges, hogy az adott ponthoz tartozó #define szerepeljen a Program.cs fájl elején (#define PART1 az 1. feladathoz, #define PART4 a 4. feladathoz, stb.).
- Csak olyan kódot tölts fel moodle-be, ami nálad fordul. Ami nálad nem fordul, a Moodle-ben sem fog.
- A fájlokat nem tömörítve kell feltölteni, hanem önmagukban (drag-and-drop-pal egyszerre be lehet húzni az összeset).
- Figyelj rá, hogy a fájlnevek pontosan azok legyenek, amiket a feladat kér!
- A Moodle által használt fordító nem feltétlenül támogatja az újabb nyelvi fejlesztéseket. Amit az órákon tanultunk és használtunk, az mind működik, de az újabb fejlesztések nem biztos.

Feladat – Kisz 2

- A feladat elkezdéséhez a mellékelt projekt tartalmaz kódokat. A megadott **Program.cs** fájl a tesztelésben segít. Módosítani csak az elején lévő #define-okon lehet, amik a teszteket aktiválják.
- A feladatban egy autóflotta kezelésének néhány leegyszerűsített funkcióját valósítjuk meg.
- Az alapkódban adott egy absztrakt **Car** osztály, ami egy autó adatait írja le. Van konstruktora, illetve egy **Consumption** absztrakt metódusa, ami az autó fogyasztását adja vissza a paraméterben kapott távolság alapján. Van továbbá egy **GetData** absztrakt metódusa, amely sztringként visszaadja az autó adatait.
- A feladatban két interfészt, valamint több osztályt kell készíteni, mindegyiket csak kevés funkcióval.
- A feladatok között vannak összefüggések.

A feladatok:

1. Hozz létre egy **IVehicle** interfészt az alábbi metódusokkal: (1 pont)
 - a. **NumberOfWheels**: paraméter nélküli, visszaadja a jármű kerekeinek számát.
 - b. **Category**: paraméter nélküli, visszaadja a jármű kategóriáját sztringként. A kategória besorolásokat a feladat végén lévő táblázat tartalmazza.
2. Hozz létre egy **PassengerCar** osztályt, amely a **Car** absztrakt osztályból származik és megvalósítja az **IVehicle** interfészt. (2 pont)
 - a. Az osztálynak legyen egy új tulajdonsága, a szállítható személyek száma (**NumberOfTransportedPersons**).
 - b. Valósítsd meg az osztály konstruktorát, hogy ezt az új tulajdonságot is onnan várja.
 - c. A **Consumption** metódus a 100 km-re vetített átlagfogyasztás és a paraméterben megkapott távolság alapján adja vissza az autó fogyasztását. (átlagfogyasztás/100 * távolság)
 - d. Valósítsa meg a **GetData** függvényt, amely a példakimenetben látható módon térjen vissza az autó adataival.
 - e. A **Category** függvény a lenti táblázat alapján adja vissza a személyautó besorolását.
 - f. A **NumberOfWheels** függvény konstansan 4-et ad vissza.
3. Hozz létre egy **FreightCar** osztályt, amely a **Car** absztrakt osztályból származik és megvalósítja az **IVehicle** interfészt. (2 pont)
 - a. Az osztálynak legyen egy új tulajdonsága, a teherbírás (**Capacity** : double).
 - b. Valósítsd meg az osztály konstruktorát, hogy ezt az új tulajdonságot is onnan várja.
 - c. A **Consumption** metódus a 100 km-re vetített átlagfogyasztás és a paraméterben megkapott távolság alapján adja vissza az autó fogyasztását. (átlagfogyasztás/100 * távolság)
 - d. Valósítsa meg a **GetData** függvényt, amely a példakimenetben látható módon térjen vissza az autó adataival.
 - e. A **Category** függvény a lenti táblázat alapján adja vissza a teherautó besorolását.
 - f. A **NumberOfWheels** függvény a teherautó besorolása alapján adja vissza a kerekeinek számát (lásd a lenti táblázatot).

4. A **Car** osztályhoz adj hozzá egy **CanGo** nem absztrakt metódust, amely egy bool értéket ad vissza attól függően, hogy az autó megteheti-e a paraméterben megadott távolságot az aktuális üzemanyag mennyiségével. (Ehhez használd az átlagfogyasztást és/vagy a **Consumption** függvényt). (2 pont)
5. Hozz létre egy új interfészt **IAvailable** néven, amely egy **Available** metódust definiál. (1 pont)
6. Az **IAvailable** interfészt implementálja a **PassengerCar** osztály. Az **Available** metódus határozza meg, hogy az autó elérhető-e vagy sem. Egy autó akkor elérhető, ha az aktuális üzemanyag mennyiségével meg tudja tenni a metódus paraméterében megadott távolságot. (2 pont)

Személyautó (ár alapján)		Kerekek száma
0 – 2.000.000	alsó kategória	4
2.000.001 – 15.000.000	közép kategória	4
15.000.001 –	felső kategória	4
Teherautó (teherbírás alapján)		
0 – 3,5t	kisteherautó	6
3,6t – 7,5t	könnyű teherautó	8
7,6t – 12t	középnehéz teherautó	12
12t –	nehéz teherautó	16