személykocsi, teherkocsi modell

Hozz létre agy Vehicle osztályt. Az általános jármű osztálynak van önsúlya, és legalább egy vezető. Ezek adják a teljes súlyát.

* Vezess be egy vehicleWeight privát attribútumot, a jármű súlya.
* PERSON\_AVERAGE\_WEIGHT konstans érték: egy személy átlagos súlyát tartalmazza. Az értéke legyen 75.
* Hozz létre egy konstruktort, amely megkap egy értéket az attribútum számára, és beállítja azt.
* Szükséges metódus a getGrossLoad visszaadja a mozgó jármű súlyát. (A jármű súlyához adjuk hozzá a sofőr súlyát.)

Hozz létre egy Car osztályt, amely az általánosabb jarmű osztályból származik (Vehicle) a következők alapján

* tartalmaz egy attribútumot
* numberPassenger: egész szám, amely az utasok számát jelenti (sofőr nélkül)
* konstruktor, amely megkap két értéket az attribútumok számára, és beállítja azokat (az ős attribútumát a superhívással).
* getGrossLoad visszaadja a mozgó gépkocsi súlyát. Hívd az ős azonos nevű metódusát, és add hozzá az utasok súlyát (Ez lesz a visszatérési érték).
* Definiáld felül a toString metódust, a következő formára:
* Car{numberOfPassenger=4, vehicleWeight=1700}

Hozz létre egy Van osztályt, amely egy kisteherautót modellez. Ez az osztály az általános autóból származik, kiterjeszti a Car osztályt.

* Tartalmaz egy cargoWeight egész attribútumot a rakomány súlyának tárolására.
* Definiálj egy konstruktort, amely megkapja a rakománysúlyt, és az ős osztály két attribútumához szükséges értékeket is (összesen három egész érték). Hívd a super-t az ős attribútumok inicilizálására, és állítsa be az új attribútumot is.
* Definiáld felül a getGrossLoad metódust. Hívd az ős azonos nevű metódusát, és add hozzá a rakomány súlyát (Ez lesz a visszatérési érték).
* Definiáld felül a toString metódust, a következő formára:
* Van{cargoWeight=1222, numberOfPassenger=4, vehicleWeight=1200}

Virtuális metódusok használata (Mindig az hívódik, amelyikre szükség van.)

A komp FerryBoat képes bármilyen autót tárolni, aminek kisebb a súlya a megengedettnél. Viszont a komp is egy speciális jármű.

* Definiálj egy állandót MAX\_CARRY\_WEIGHT néven, amely tárolja a maximálisan szállítható autó súlyát.
* Ha szállít autót, akkor azt egy Car típusú car attribútumba tárold. (Ha nem szállít, akkor ez null.)
* Hozz létre egy konstruktort, amely megkap egy egész értéket, amivel meghívod super-t.
* Definiáld felül a getGrossLoad metódust. Hívd az ős azonos nevű metódusát, és add hozzá a szállított autó súlyát (Ez lesz a visszatérési érték).
* Definiálj egy canCarry metódust, ami igazat ad vissza, ha a paraméterben kapott autót szállíthatja, azaz a szállítandó autó súlya kisebb, mint a MAX\_CARRY\_WEIGHT (egyébként hamis). (Akár autó, akár kisbusz a referencia célja mindig a megfelelő metódus hívódik meg.)
* Definiálj egy load metódust, ami igazat ad vissza, ha a paraméterben kapott autót berakodta, azaz a súlya kisebb, mint a MAX\_CARRY\_WEIGHT (ekkor tárolja el az autót a referenciába).
* Definiáld felül a toString metódust, a következő formára FerryBoat= + a tárolt autó toString eredménye:
* FerryBoat{car=Van{cargoWeight=200, numberOfPassenger=1, vehicleWeight=1200}}
* FerryBoat{car=Car{cargoWeight=200, numberOfPassenger=1, vehicleWeight=1200}}