

TUDNIVALÓK: Lévén, hogy nagyon kevés időnk van a konzultációkon, a feladatok zöme HÁZI FELADAT !

Ezek megoldása fontos a tárgy sikeres teljesítéséhez! (Nyilván minden feladatsorban vannak hasonló jellegű feladatok, ki-ki maga tudja, hogy mennyire boldogul velük, ha könnyen, akkor természetesen nem kell megoldani az összes „egyforma” feladatot.)



1. feladat



Folytassuk a múlt órai feladatot, és bővítsük ki néhány további funkcióval.

- Írjuk meg az elmaradt PTEsResztvevo osztályt!
 - Módosítsuk az adatbeolvasást úgy, hogy egyrészt később könnyen lehessen módosítani, másrészt jar fájlból futtatva is működjön.
 - Végezzük el a szimulációt.
- Határozzuk meg a maximális bevételt
 - Írassuk ki a kedvezményezettek azonosítóját és azt, hogy fejenként összesen hány Ft kedvezményt kaptak
 - Rendezzük különböző szempontok szerint a rendezvénylistát.
 - Teszteljük az elkészült osztályokat (vagy azok egy részét).

(Hf-ként próbáljon ki még egy-két teszt-esetet.)

Segítség:

<http://oopbook.com/junit-testing/junit-testing-in-netbeans/>

<http://tutorials.jenkov.com/java-unit-testing/simple-test.html> (általában is érdemes olvasgatni a Jenkov-tutoriálokat),

otthonra: <https://www.youtube.com/watch?v=Q0ue-T0Z6Zs>

A feladatok megoldása előtt célszerű – legalább vázlatosan – elkészíteni az osztálydiagramot.

2. feladat

Folytassuk az előző feladatsor második feladatát, onnan, hogy a diákok felvették a tantárgyakat. Emlékeztető:

A diákokat a nevük, kódjuk és az életkoruk jellemzi. A tantárgyak nevükkel, kódjukkal és kredit-számukkal adhatók meg. Tantárgyfelvételkor az adott tantárgy bekerül a diák felvett tantárgyainak listájába, a felvett kreditek száma pedig értelemszerűen növekszik.

(ld. gyak2_indulo projekt)

a) Módosítsuk néhány aprósággal a projektet:

- A kommentben kért módosítás;
- Tegyük lehetővé, hogy a tantárgyfelvétel után láthassuk a diákok kreditszámát is;
- Beolvasás után a lehetséges tantárgyakat is írassuk ki (lehetőleg új metódus írása nélkül, a régi átírásával).
- Szervezzük külön csomagokba az osztályokat, illetve vegyük külön a beolvasást.

b) Vannak költségtérítéses diákok is. Nekik fizetniük kell a felvett kreditek után, mégpedig a kreditszám valahányszorosát. Regisztrálásukkor meg kell adni a fizető felet is (cég vagy saját maga). Írassuk ki a névsorukat, és állapítsuk meg, hogy összesen mennyit kellett fizetniük.

c) Rendezzük a diákokat különböző szempontok szerint!

d) Oldjuk meg, hogy az adatfájlok is a jar-ba kerüljenek!

Továbbfejlesztés:

- Oldjuk meg, hogy a diák vizsgázhasson is, és szimuláljuk a diákok „vizsgáztatását”. Vizsgázás során az adott tárgy kredit-értékével növekszik a teljesített kreditek száma. (Most minden vizsgát sikeresnek tekintünk, ☺) A szimuláció azt jelenti, hogy valahányszor lefuttatjuk ezt: egy véletlenül kiválasztott diák vizsgázik egy véletlenül kiválasztott tárgyból.
- Csak sikeres vizsga esetén növekedjen a kreditszám. Számolja ki a diák átlagát is. (A vizsgajegy megadható paraméterként.)
A szimuláció során a jegyeket is véletlenszerűen generálja.
- Keresse meg, hogy kik teljesítették a legtöbb kreditet, kiknek a legrosszabb az átlaguk, kik a legfiatalabbak, stb.

3. feladat



Folytassa az előző feladatsor feladatát! Eddig ennyi volt:
A JAVA-TRAVEL utazási iroda több hajót működtet, ezeken utaztatja a vendégeit.

Minden hajóút esetén adott annak neve és egy egyedi azonosítója, ezen kívül pedig az utaztatható személyek maximális száma. A hajóra akkor tud felszállni egy utas, ha az utaslétszám még ezen a korláton belül van, és az utas beszállhat. Ha beszállhat, akkor az utas bekerül a hajó utas-

listájába. Egy-egy út költsége egy-egy adott hajón minden utas számára azonos.

Az utasokat a nevük és egy egyedi kód azonosítja. Mindenkinek van valamennyi pénze, és bármikor tud költeni valamennyit és kapni is valamennyit. A társaság csak akkor enged beszállni valakit, ha ki tudja fizetni az adott út költségét, plusz még ezen felül marad nála egy minden utas számára egyformán kötelező alaptőke.

Írjunk programot az iroda működtetésére, vagyis olvassuk be néhány hajó és néhány utas adatait, szimuláljuk az utazást, majd írassuk ki az adatokat.

A szimuláció ezt jelenti: minden hajó esetén adjuk meg az illető hajóra érvényes útiköltséget (bizonyos határok között véletlenül generált érték), majd valahányszor egymás után válasszunk ki egy véletlen hajót, amelyre egy véletlen utas megpróbál felszállni.

Folytatás:

Bővítse az eddigieket:

- A cég kedvezményt ad a Java programozók számára ☺, de mivel nem mindenki egyformán jó programozó, ezért a kedvezmény százalékát egyéneként dönti el. (A Java programozók azonban reménykednek abban, hogy esetleg a családtagjuk, barátjuk is kedvezményt kaphat, ezért úgy írják meg a programot, hogy csak az legyen érdekes, hogy valaki kedvezményezett. ☺) Kiíratáskor az is kerüljön a kedvezményezett neve mellé, hogy hány százalék kedvezményt kap.
- További feladatként számoljuk ki a cég teljes bevételét. Gondolja végig, hogy ehhez mit, melyik osztályban és hogyan kell megadni.

További feladatok a „szokásosak”: melyik hajónak legtöbb a bevétele, melyiken utaznak a legtöbbben/legkevesebben, valamilyen szempont szerinti rendezés, stb.

3. b – kreatív feladat

Az eredeti feladat szerint csak mese az, hogy hajóról beszélünk, hiszen semmi olyan nincs benne, ami valóban csak a hajókra lenne jellemző.

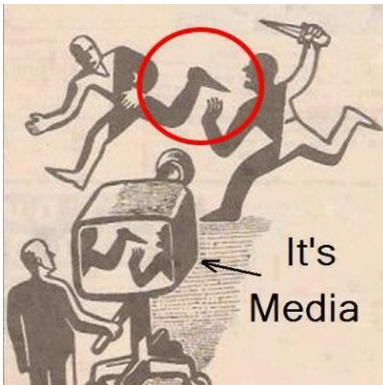
Találjon ki egy olyan Hajo osztályt, amelynek tulajdonságai, metódusai (nem kell sok) valóban a hajókra jellemző.

Találjon ki egy olyan SpecialisHajo osztályt is, amely öröklődik az eredeti Hajo osztályból, de persze, van benne valami saját specialitás is.



Ezt a két osztályt foglalja bele egy közös programba (ha akarja, kitalálhat további osztályokat is). Nem kell túl bonyolult programot írni, de mozgassa meg egy kicsit a fantáziáját.

4. feladat



Ez is folytatása a múltkorinak. Eddig ennyi volt:

Egy **újság**ot egyértelműen jellemez a *neve* és *megjelenési dátuma* (sima String). Az újság *cikket közöl()*, vagyis a paraméterében megadott cikket hozzáadja az újságban megjelent *cikkek listájához*.

Egy **cikk** egyértelműen megadható a *szerző nevével*, a *cikk címével*, a *cikk méretével* (karakterszám) és egy, a cikkben lévő hazugság nagyságára utaló *százalékláb* értékkel. A cikk címe szükség esetén megváltozhat, egyéb jellemzői nem.

Az újságban a cikk közlésekor azt is számolják, hogy a paraméterben lévő cikk mennyivel változtatja meg az *átlagos hazugságszázalékot*. Ha még egyetlen cikket sem közöltek, akkor az átlagszázalék lekérdezésekor -1-et adjon vissza.

Olvassuk be valahány újság és cikk adatait (*cikkek.txt* és *ujsgok.txt* fájl), majd az *ujsgIras()* során véletlen sokszor egy-egy véletlenül választott újságban jelentessünk meg egy-egy véletlenül választott cikket. Végül írassuk ki az újságok adatait, és számoljuk ki, hogy mekkora volt az újságok átlaghazugsága (az egyes újságok átlaghazugságának átlaga). Számolja ki, hogy összesen hány cikk jelent meg, illetve azt is, hogy melyik a leghazugabb újság.

Szorgalmi: A megjelenés dátumát ne String-ként, hanem rendes dátumként kezelje. Segítség a múltkori feladatsor végén, vagy ha előre akar szaladni, akkor itt:

<http://javarevisited.blogspot.hu/2015/03/20-examples-of-date-and-time-api-from-Java8.html> .

Folytatás:

Tovább lépve a következőt tudjuk még:

A médiabirodalom nyomtatott és internetes újságot is megjelentet.

A **nyomtatott újság**ot a néven és megjelenési dátumán kívül jellemzi még az újság *példányszáma* és a *mérete* (összesen hány karaktert tud megjelentetni). Mivel itt korlátozott a méret, ezért egy cikk közlése csak akkor lehetséges, ha annak mérete még belefér az újság méretébe. Egy cikk közlésekor természetesen csökkenteni kell a rendelkezésre álló *maradék méretet*.

A nyomtatott újság *olvasóinak száma* akkor növekszik, ha *elad()*ják az újság egy példányát. Nyilván csak a példányszám erejéig.



Az **internetes újság** a néven és megjelenési dátumon kívül tartalmazza még az újság *linkjét* (most csak egy String). Olvasóinak száma *kattintás()*kor növekszik.

Egyelőre ennyi, de esetleg végiggondolhat még néhány, az internetes bulvárra jellemző dolgot, pl.:

1. Előfordulhat, hogy egy-egy cikk esetén *címmódosítás()*ra van szükség, mégpedig akkor, ha a vizsgálat pillanatában az *olvasók száma* nem haladott meg egy bizonyos létszámot.

2. Egy-egy cikkhez *fórumbejegyzés()* is születhet. Jó lenne figyelni a *fórumozók számát*, de akár magukat a megjegyzéseket is.

a/ Alakítsa át a korábban megírt adatbevitelt úgy, hogy az *újságok_orokleshez.txt* fájlból olvasson – ebben értelemszerűen látszik, hogy melyik internetes, melyik nyomtatott, és mit jelentenek a sorokban szereplő adatok.

b/ Válogassa szét az internetes és a nyomtatott újságokat, és külön-külön írassa ki őket.

c/ Bővítse ki a feladatot, és „terjessze” az újságokat. Ez azt jelenti, hogy néhányszor (véletlen sokszor) válasszunk ki véletlenszerűen egy-egy újságot, és növeljük olvasóinak számát. Utána írassuk ki az újságok nevét a hozzájuk tartozó olvasószámmal együtt. Az internetes vagy a nyomtatott újságoknak van nagyobb olvasóközönsége?

d/ Az újságokat olvasószám szerint rendezve írassa ki. Esetleg próbáljon ki más rendezési szempontot is.

e/ Alakítsa át az ösosztályt absztrakt osztállyá.

f/ Írja meg a feladat megoldásának alapját szolgáló interfészt.

g/ Bővítse a feladatot a korábban említett internetes bulvárra jellemző tulajdonságokkal.

5. feladat – mielőtt hiányolná az italokat ☺



Az előző feladatok sikeres megoldása után „ihat” egyet, és folytathatja a múltkori feladatsor ide vonatkozó feladatát.

Az italokat jellemzi az ital fajtája (pl. tej, bor, stb.), vonalkódja, mennyisége (literben), literenkénti egységára (Ft/liter) és a tényleges eladási ár, amely már az ÁFA értékét is tartalmazza.

Vannak emberek, akik szeretnek inni. ☺ Ezeket az embereket jellemzi a nevük, és van valamennyi pénzük. Elvileg fizetnek az italért. Mivel többször is rendel(het)nek, ezért ilyenkor a fizetendő összeg az aktuálisan fogyasztott ital árával növekszik.

Az ember egy szöveges számlát „kap”, erre vagy a fizetendő összeg kerül, ha azt ki tudja fizetni, vagy az, hogy a különbözet (azaz a fizetendő és a pénz közötti különbség) értékben köteles mosogatni. Ekkor „büntetésképpen” és az adminisztrálhatóság kedvéért a neve is rákerül a számlára

Az ember fizet(), ha tud, azaz levonódik az illető pénzéből a fizetendő érték, de a maradék nem mehet át negatívba (hiszen „le mosogatja” a tartozást).

Játsszon kicsit a feladattal, vagyis hozzon létre italokból és/vagy emberekből álló listá(ka)t, és pl. az összes ember rendelhet valamilyen italt saját számlára, de akár úgy is, hogy egyikőjük fizeti a teljes összeget, vagy próbálkozhat avval, hogy egyetlen ember „végigkóstolja” a teljes italválasztékot, stb. Működtesse a fantáziáját.

Folytatás:

Hozzunk létre egy AlkoholosItal osztályt. Az alkoholos ital is Ital, de az ital jellemzőin kívül jellemzi még az ital márkája és alkoholfoka. Ezek nem is változhatnak meg az ital élettartama alatt. Megállapítandó, hogy az adott italmennyiség ártalmas-e már: akkor ártalmas, ha a benne lévő alkohol mennyisége több, mint egy általánosan rögzített alkoholhatár. (megart())

Egészítsük ki, illetve módosítsuk az Ember osztályt
Csak akkor ihat alkoholt, ha elég idős (azaz egységesen adott korhatárnál idősebb), és a rendelt italmennyiség nem árt meg.



Szorgalmi: Az aktuális év megadására használja a rendszeridőt (ehhez az aktuális calendar példány évszámát kell lekérni.)

Finomítsuk a „megártás” fogalmát, és egészítsük ki a feladatot avval, hogy valaki csak akkor ihat alkoholt, ha eleget tesz a korábbi feltételeknek (elég idős és nem árt meg az ital), és még a véralkoholszintje nem ér el egy, a törvényben rögzített határt. (A törvény nem tesz különbséget a nemek között, de a véralkoholszint kiszámításához szükséges faktor eltérő férfiak és nők esetén – de mindkét nemre egy-egy általánosan elfogadott érték jellemző.)

A véralkoholszint kiszámítási módja:

$$\text{véralkoholszint} = \text{fogyasztott_alkoholmennyiség} / \text{testsúly_kg} / \text{Widmark_faktor}$$

(A kapott érték ezreléket jelent, vagyis az olvashatóság kedvéért még szorozni kell 1000-rel.)

Ha már a véralkoholszint miatt meg kell különböztetnünk egymástól a férfiakat és a nőket, legyen még egy különbség: férfiak esetén figyeljük azt is, hogy nős-e az illető. Nős férfiak esetén egy metódus mondja meg, hogy sodrófával várják-e otthon, vagy sem – akkor jár sodrófa, ha az illető véralkoholszintje meghalad egy bizonyos – az otthoni toleranciától függő – határt.

Hol lehet beállítani a fogyasztott alkoholmennyiség értékét? Nyilván akkor, amikor iszik. Feltehetjük, hogy akkor iszik, amikor megrendeli az italt. Vagyis módosítsuk a rendel() metódust a következők szerint:

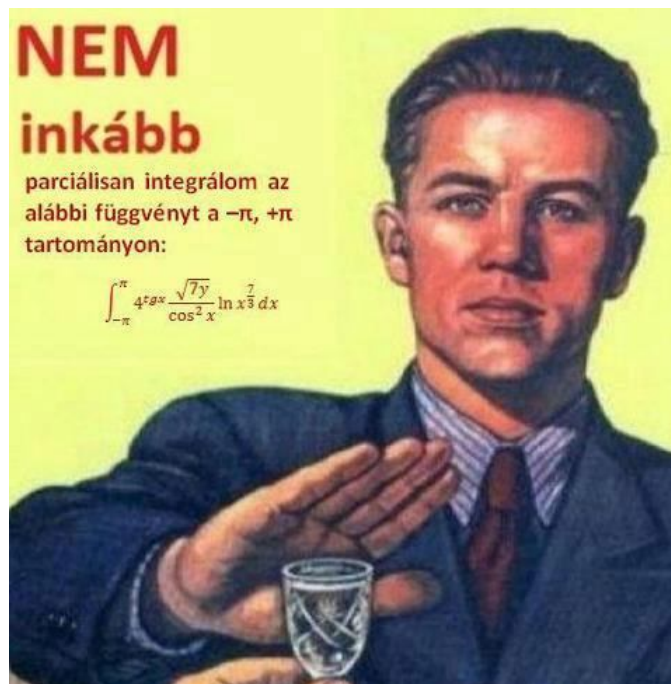
Ha ihat, akkor növelje a fizetendő értéket az ital árával. Ha az ital alkohol, akkor a fogyasztott alkoholmennyiség értékét is növelje a fogyasztásnak megfelelően. (a rendelt ital alkoholmennyiségével)

Módosítsa úgy az előző metódust, hogy a vezérlő osztály azt is ki tudja írni, hogy az illető nem ihat, ezért nem is rendelhet. (Segítség: gondolja végig a metódus típusát.)

Adjon rá lehetőséget, hogy az illető többször is bulizhasson, azaz írjon egy ujBuli() metódust, amelyben kinullázza a szükséges kezdőértékeket.

Rendezzen mindenféle szempont szerint.

Ha ezt mind-mind jól megoldotta, akkor jutalomból ihat egy igazi pohárral is a kedvencéből, de vigyázzon, meg ne ártson! ☺



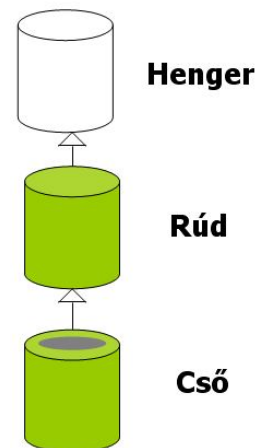
FUTOTTAK MÉG KATEGÓRIA (Vagyis egyszerűbb gyakorló feladatok)

6. feladat

Ha esetleg még nem találkozott vele ☺.

Különböző henger alakú testek nyilvántartását végző programot kell írni:

- Vannak olyan **hengerek**, amelyek csupán mértani testek. Ezeknek van sugaruk és magasságuk.
- Vannak **rudak**, melyeknek tömegük is van (tömör hengerek). Az ilyen alakzatoknak van sűrűségük is. Ha nem adunk meg sűrűség értéket, akkor annak értéke egységnyi (1 g/cm³). (Vajon milyen anyagból készülhetett? ☺)
- Végül vannak **csövek**: valamekkora falvastagságú tömör, lyukas hengerek



a/ Készítsük el a megadott alakzatok osztályait, figyelembe véve az alakzatok közötti hierarchikus kapcsolatot (öröklődést).

Minden alakzat legyen képes visszaadni alapadatait, térfogatát és esetleges tömegét. (toString())

Megjegyzések:

- A henger térfogata: $\text{sugár} \cdot \text{sugár} \cdot \text{PI} \cdot \text{magasság}$ ☺
- A rúd tömege: $\text{sűrűség} \cdot \text{térfogat}$
- A rúd térfogata: egyenlő a henger térfogatával
- A cső térfogata: van egy külső térfogat, amely a teljes henger térfogata és egy belső térfogat, amely a belső (nem tömör) rész térfogata. A cső térfogata a külső és a belső térfogatok különbsége.

A „belső” henger egy olyan henger, amelynek a sugara a külső henger sugara mínusz a falvastagság. → a belső henger egy új objektum lesz.

A cső megadható a külső/belső henger sugarának segítségével is.

Figyelem! Bár egyszerű lenne, de ne találjon ki saját képletet a cső térfogatának kiszámítására, hanem az öröklődést figyelembe véve az ős térfogata alapján számoljon!

b/ Készítsünk indító osztályt, amelyben létrehozunk egy hengert, két rudat (egy egységnyi és egy ettől eltérő sűrűségűt) és két csövet (egy egységnyi és egy ettől eltérő sűrűségűt), és a program kiírja az objektumok megfelelő adatait. (A kiíratást feltétlenül egészítse ki **mértékegységekkel**. Anélkül semmit sem ér egy odabökött szám.)

HF (szorgalmi): Nézzzen utána a help-ben, hogy

(1) hogyan lehetne lekérni az osztály nevét (this. valami ☺)

(2) hogyan lehetne beállítani, hogy csak valahány tizedes jegyet írjon ki a toString()-ben megadott kiszámított érték esetén is. (NumberFormat, ill. string format).

Előbb **készítsük el a feladat UML ábráját**, majd a Java megvalósítást!

7. feladat

Egy vállalatban vannak dolgozók és főnökök.

A **dolgozót** a következők jellemzik: név, személyi szám, fizetés, életkor.

A fizetése az iskolai végzettségétől és az életkorától függ, mégpedig így:

– Iskolai végzettségtől függően három kategória lehet: 1,2 vagy 3.

(1: általános, 2: közép, 3: felsőfokú végzettség)

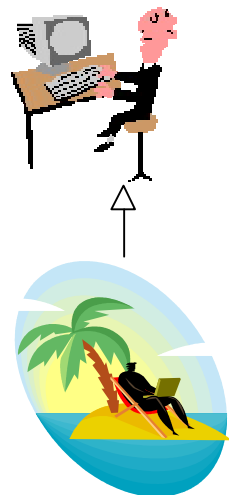
– Van egy mindenkire jellemző alpbér, amely elvileg időnként változhat.

– Fizetési kategóriától függően a fizetés az alpbér X-, Y-, vagy Z-szerese..

– A dolgozók kategóriától függetlenül korpótlékot is kapnak, mégpedig éveik számának Kszorosát.

A **főnök** is dolgozó, csak őt a fentiek mellett jellemzi még a

- beosztottak száma,
- kinevezésekor azonnal harmadik kategóriájú lesz, és ez soha többé nem csökken.
- fizetése pedig kívül a beosztottak számának K-szorosával.



Készítse el a feladat-specifikációnak megfelelő **UML osztálydiagramokat** és az azt megvalósító Java osztályokat (konstruktorokkal). A megoldáshoz felhasználhatja a házi feladatként megírt Dolgozo osztályt.

A fenti két osztály felhasználásával oldja meg a következő feladatot:

- Hozzon létre egy főnök objektumot!
- Kérdezze meg, hogy hány dolgozót vesznek fel, majd gondoskodjon ennyi dolgozó felvételéről! (Kérje be az adataikat és tárolja őket).
- Írassa ki a cég dolgozóinak adatait! (A főnökét is.)
- A főnök adatainak kiíratása után írassa ki fizetésük szerint csökkenő sorrendben a dolgozók nevét, fizetését.

A szükséges osztályváltozók értékét generálja véletlen számként.

7.b feladat

A dolgozókat ne sorolja fixen három kategóriába, hanem írjon meg egy BerTabla osztályt, amelyben megadhatunk egy kategória – szorzó táblázatot a hozzá tartozó alpbérrel, ezen kívül pl. nyelvenkénti nyelvvizsgapótlékot, gyerekenkénti családi pótlékot, stb. És a dolgozó fizetését ez alapján számoljuk.

Szorgalmi: Próbálja meg úgy módosítani a feladatot, hogy a születési dátumot ne kelljen külön megadni, hanem kinyerhető legyen a személyi számból.

8. feladat

Egy **könyvet** jellemez a szerzője, címe, oldalszáma, ára. Az ár az oldalszám valahányszorosa. A könyv ki tudja írni a saját jellemzőit.

Egy **tankönyvet** szintén jellemez a szerzője, címe, oldalszáma, ára, de ezen felül még az is, hogy melyik tárgy ajánlott irodalma. A tankönyvek árából lejön a jegyzettámogatás, amely az ár x százaléka. Ő is ki tudja írni a saját jellemzőit.

Az **idegen nyelvű könyvet** szintén jellemzi a szerzője, címe, oldalszáma, ára, de ezen felül még az is, hogy milyen nyelven íródott, és egy nehézségi szint: A, K, F (alap, középfokú, felsőfokú), amely azt jelzi, hogy milyen nehéz az idegen nyelvű szöveg. Az ő ára is az oldalszám valahányszorosa, de nem biztos, hogy ugyanaz a szorzó, mint egy sima könyvnél. És ő is ki tudja írni a saját jellemzőit.

- Írassa ki legalább egy regény és egy tankönyv adatait!
- Olvassa be n db idegen nyelvű könyv adatait, majd kérjen be egy idegen nyelvet, és írassa ki az azon a nyelven olvasható könyvek listáját.

Készítsen osztálydiagramot!

Szorgalmi: Gondolja végig, és oldja meg úgy a feladatot, hogy legyen még egy negyedik könyvtípus is: **idegen nyelvű tankönyv**, amely egyszerre tankönyv is és idegen nyelvű könyv is. A vezérlő osztályban hozzon létre egy-egy példányt mind a négy fajtából!



9. feladat

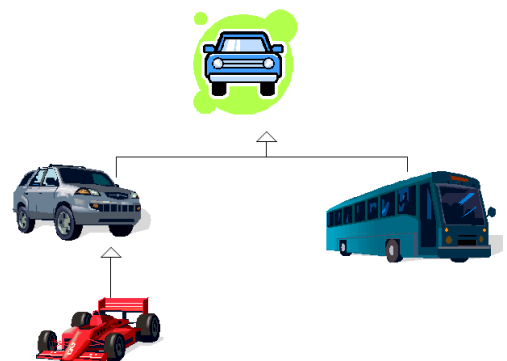
Feladatunk a gépjárművekkel foglalkozik.

Egy **gépjármű** jellemző tulajdonságai:

- színe, rendszáma, üzemanyagszintje, fogyasztása, stb.
- indítás(), gyorsítás(), fékezés(), tankolás(), stb.

Feladatok:

- Készítse el a GepJarmu osztályt.
- Készítse el a SzemelyGepKocsi, Busz és SportKocsi osztályokat (értelmes változókkal és metódusokkal).
- Készítsen főprogramot (legyen kreatív).



10. feladat

Találjon ki egy feladatot és oldja meg.

A feladatban legyen

- öröklődés
- adattagok, metódusok
- listák
- lista-elemeken végzett műveletek, rendezés, stb.
- fájlból való olvasás
- esetleg véletlen szám generálás