

TUDNIVALÓK:

Most is és a következő gyakorlatokon is – akkor is, ha külön nem emeljük ki – az órán meg nem oldott feladatok HÁZI FELADAT-ként megoldandóak!!!

Ez fontos a tárgy sikeres teljesítéséhez!

(Némelyik feladathoz segítő ötleteket talál a feladatsor végén.)

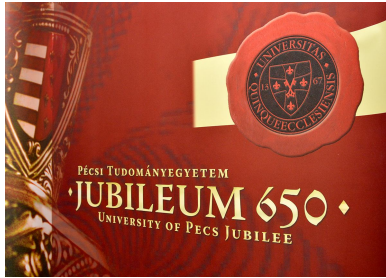


Megjegyzés: Ha valaki Eclipse-ben szeretne dolgozni, akkor úgy tűnik, hogy ez alapján a link alapján úgy is le lehet tölteni, hogy tudja kezelni a Swing csomagot:

<http://www.rcp-vision.com/?p=1132&lang=en>

Természetesen egyáltalán nem kizárt, hogy más helyről letöltve is megoldható (pl. <http://website-box.net/se-keyword/eclipse+java+ui+designer>). Ha valaki ezt választja, és zh előtt legalább egy héttel szól, akkor reményeim szerint megoldható, hogy a laborgépen is fusson ez a változat.

1. feladat



Folytassuk a múlt órai feladatot, és teszteljük az elkészült programot. (Ténylegesen már jóval korábban kellett volna, de nem fér bele minden egy gyakorlatba.)

(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-tutoriókat),

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

Bővítsük ki a megoldást néhány további funkcióval:

a/ Állapítsuk meg, melyik rendezvénynek volt legnagyobb a bevétele!

b/ Írassuk ki a kedvezményezettek azonosítóját és azt, hogy fejenként összesen hány Ft kedvezményt kaptak.

c/ Írassuk ki a rendezvényeket résztvevőszám szerint csökkenő sorrendben.

d/ Írassuk ki a rendezvényeket egy táblázatba:

Cím	Időpont	Jegyár (Ft)	Résztvevőszám	Bevétel (Ft)
Pítherapy rapper duó	szept.2.	800	10	7680
Carmina Burana	aug.31.	1000	7	6800
Középkori egyetemi kiállítás	szept.1.	700	7	4620
PTE Táncegyüttes	szept.2.	500	7	3350
Halott Pénz	szept.2.	2000	3	6000

TOVÁBBI FELADATOK:

A megoldott HF-ek mindegyikéhez készítsen teszt-eseteket is.

2. feladat



Ha már megoldotta az előző feladatsorban szereplő JAVA-TRAVEL feladatot, akkor már csak a tesztelés van hátra. Tesztelje a megoldásban szereplő osztályokat.

Ha esetleg nem oldotta meg, akkor oldja meg és tesztelje.

További feladatok a „szokásosak”: melyik hajónak legtöbb a bevétele, melyiken utaznak a legtöbben/legkevesebben, stb.

Próbálja meg ilyen formában is kiírni az eredményt:

Hajónév	Azonosító	Utiköltség	Utasszám
Pearl of Java	PJ001	1011	10
Queen of Java	QJ002	6533	9
King of Java	KJ003	12605	8

2. 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.

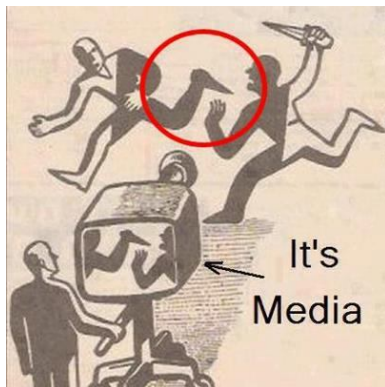
Találjon ki egy olyan Hajo osztályt, amelynek tulajdonságai, metódusai (nem kell sok) valóban a hajókra jellemző.

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.



3. feladat – a médiáról



A feladatmegoldáshoz készítsen UML ábrát is.

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 hazugságsszázalékot*. Ha még egyetlen cikket sem közöltek, akkor az átlagszázalék lekérdezésekor -1-et adjon vissza.

Olvassuk be valahány újság és cikk adatait, majd az *ujsglras()* 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.

A cikkek és újságok adatait a *cikkek.txt* és az *ujsgok.txt* fájlból olvassuk be.

A megoldáshoz készítsen teszt osztály(oka)t is.

Szorgalmi: A megjelenés dátumát ne String-ként, hanem rendes dátumként kezelje. Segítség az előadás anyagában, illetve a 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> .

3. b feladat

Folytassa az előző feladatot. 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 *ujzagok2.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 öröklődést is tesztelje.

e/ Alakítsa át az őosztályt absztrakt osztállyá.

f/ Írja meg a feladat megoldásának alapját szolgáló interfészt.

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

h/ **Szorgalmi:** Az újságokat olvasószám szerint rendezve írassa ki.

4. feladat – mielőtt hiányolná az italokat ☺



Az előző feladatok sikeres megoldása után „ihat” egyet.

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()hetnek, 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 pénze 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 „lemosogatja” 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.

4. b feladat

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())

Teszt: állítsunk be egy alkoholhatárt, és írassuk ki, hogy az egyik adott ital megárt-e.

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 – ld. a 2. feladat szorgalmi részéhez fűzött segítséget.)

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:

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

(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 nő-s-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)

Teszt: állítsunk be a szükséges statikus adatokat, adjuk meg az egyik ember testsúlyát, esetleg módosítsuk a fogyasztandó ital mennyiségét, az ember rendeljen, majd írassuk ki az ivott alkohol mennyiségét és a véralkoholszintet.

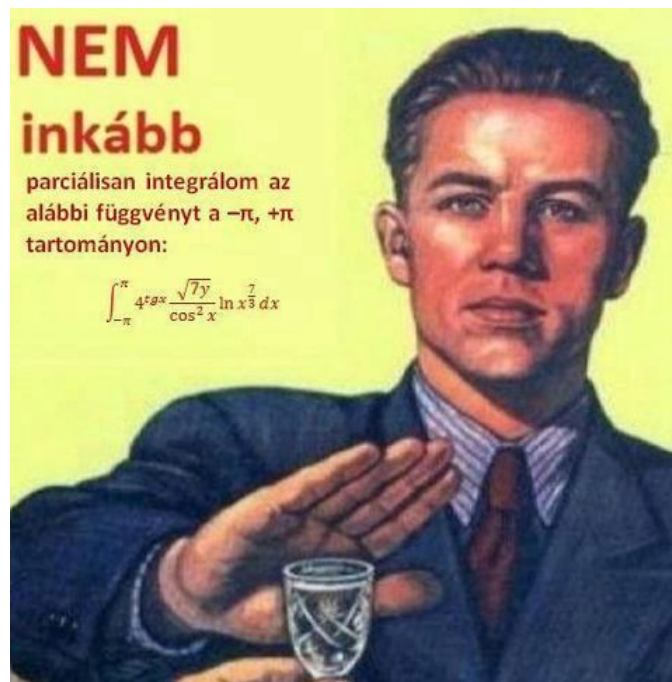
Módosítsa úgy az előző metódust, hogy a vezérlő osztály azt is ki tudja írni, hogy az illető nem ihat, ezért nem is rendelhet. (Segítség: gondolja végig a metódus típusát.)

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

Írjon toString()-et az új osztályokhoz is.

Próbáljon ki különböző teszt-helyzeteket, akár úgy is, hogy tömböket vagy listákat használ.

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



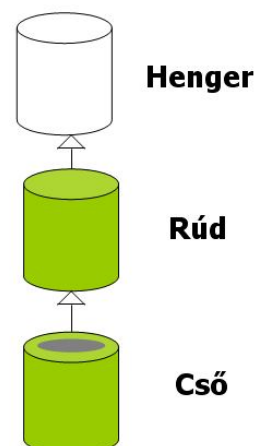
FUTOTTAK MÉG KATEGÓRIA (Vagyis egyszerűbb gyakorló feladatok)

5. feladat

Ha esetleg még nem találkozott vele ☺.

Különböző henger alakú testek nyilvántartását végző programot kell írni:

- Van olyan **hengerek**, amelyek csupán mértani testek. Ezeknek van sugaruk és magasságuk.
- Van **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): Nézzon 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!

6. 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)

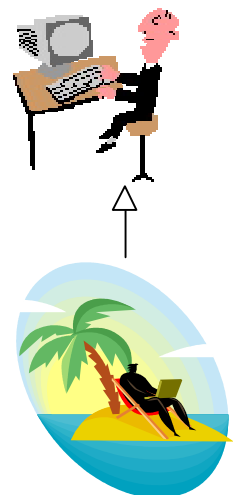
– Van egy mindenkire jellemző alaphér, amely elvileg időnként változhat.

– Fizetési kategóriától függően a fizetés az alaphér 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 kibővül 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 kiírása után írassa ki fizetésük szerint csökkenő sorrendben a dolgozók nevét, fizetését.

A szükséges osztályváltozók értékét generálja véletlen számként.

6.b feladat

A dolgozókat ne sorolja fixen három kategóriába, hanem írjon meg egy BerTabla osztályt, amelyben megadhatunk egy kategória – szorzó táblázatot a hozzá tartozó alaphérrel, 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.

7. feladat

Egy **könyvet** jellemez a szerzője, címe, oldalszáma, ára. Az ár az oldalszám valahányszorososa. 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 valahányszorososa, 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!



8. feladat

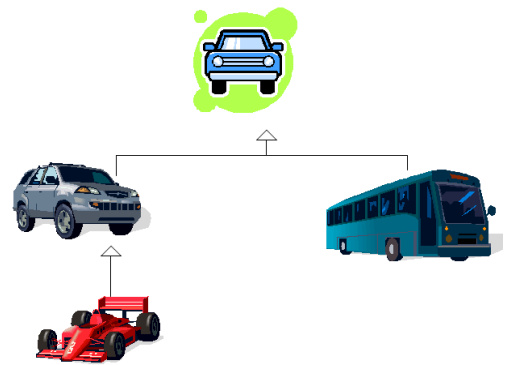
Feladatunk a gépjárművekkel foglalkozik.

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).



9. feladat

Készítsen osztályt a Caesar-titkosításhoz! (A Caesar-titkosítás során a karakterek ASCII kódjaihoz egy egész számot adunk, így kapunk egy új karaktert. Az ABC-t körbe-körbe kezeli, tehát a Z után újra az A jön.)

Az osztálynak két metódusa van, egy kodol() és egy dekodol(). Az alap ABC az angol ABC kisbetűi. Az egyéb karaktereket (pl. szóköz, számjegyek) az egyszerűség kedvéért nem változtatjuk meg.

Segítség a feladatok megoldásához:

3. feladat szorgalmi részéhez:

A dátum kezeléséhez a `Date` (`java.util` csomag) és a `Calendar` (`java.util` csomag) osztályokra lesz szükség. (Meg lehet oldani másképp is, de talán ez a legegyszerűbb.)

`Calendar` típusú példányt `Calendar.getInstance()` módon lehet létrehozni (hogypontosan miért, és ez pontosan mit jelent, arról lesz majd szó később).

A `Date` típusú dátumot a `setTime()` metódussal lehet átadni a `Calendar` példánynak.

Ettől kezdve a `calendar` példány `get()` metódusával vissza lehet adni a megadott dátumhoz tartozó évszámot, hónapot, napot, stb.

pl. `calendar.get(Calendar.YEAR)` – a nap: `calendar.get(Calendar.DATE)`.

Az `Ujsag` osztályban célszerű megírni egy `String getDateAsString()` metódust, amely a megadott dátum string alakját adja vissza. (Ebben kell alkalmazni a fent leírtakat.)

A dátum beolvasásakor a dátumot jelentő stringet is fel kell vágni. A pont speciális karakter, arra így lehet hivatkozni: `split("\\.")`;

A szétvágott dátum-string elemeit integerré kell konvertálni (`Integer.parseInt()` metódus). Az így kapott év, hó, nap értékeket át lehet adni a `calendar` példánynak egy `set()` metódus segítségével, majd a `calendar` példánytól a `getTime()` metódus segítségével le lehet kérni a `Date` típusú értéket.

(Gyorsabb megoldani, mint ezt végigolvasni, de egyelőre pont azért szorgalmi, hogy próbálja kitalálni. Később majd megbeszéljük.)

6. feladat szorgalmi részéhez:

A személyi szám egy 13 karakterből álló `String`, amelynek 2. és 9. karaktere egy kötőjel, a 3., 4., 5., 6. karakter pedig a születési év számai. (Az igaziban nem pont így van, ha kedve van, nézze meg a dátumkezelést, és annak megfelelően alakítsa át a megoldást.)

(Az ellenőrzéshez használja a `String` osztály `length()`, `charAt()` és `substring()` metódusát – ld. HELP.)

9. feladathoz:

Gondolja végig, hogy hol adjuk meg a titkosításhoz használt számot (paraméterezés), illetve azt is, hogy változhat-e példányonként a kódolás/dekódolás módja (vagyis példány- vagy osztálymetódus).

A megoldáshoz célszerű a `String` osztály megfelelő metódusait használni – ld. HELP.