

1. Modellezzük egy nagyváros **tömegközlekedését** biztosító járműparkot, és annak szervizelését!

Az önkormányzat a vele előzetesen szerződött cégeknél végezteti a szervizelést. Egy járműnek ismert az azonosítója (sztring), fajtája (villamos, autóbusz, trolibusz), gyártási éve (gyév), használati öveze (belváros, külváros, vegyes), újkori ára (újár). A jármű aktuális értékét az alábbi képlet alapján számolhatjuk: $\text{újár} \cdot (100 - (\text{jelen} - \text{év})) / (100.0 \cdot \text{faktor})$ (egészre kerekítve), ahol a faktor a jármű fajtájától és a használati övezetétől függő pozitív valós szám.

Faktor	belváros	külváros	vegyes
villamos	1.0	0.9	1.2
autóbusz	2.0	2.0	2.5
trolibusz	3.0	3.1	3.8

Egy szervizelés munkalapja tartalmazza, hogy mikor, melyik cég, melyik jármű szervizelését végezte, mettől meddig, és mekkora összegért. Az ár és a befejezési időpont csak a már lezárt szervizelések esetén ismert. A szervizelés lehet időszakos átvizsgálás vagy javítás.

- Mennyire elöregedett az autóbusz park? (15 évnél régebbi buszok és az összes busz számainak aránya)
- Melyik a legnagyobb értékű jármű (a jelenlegi érték szerint)?
- Mekkora a javítás alatt álló járművek aránya az összes járműhöz képest (időszakos átvizsgálást nem számítva)?
- Melyik jármű szervizelésére költötte az önkormányzat a legtöbb pénzt 2022-ben?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely megmutatja egy önkormányzat egyik belvárosban közlekedő villamosát, és két, egy belvárosi és egy külvárosi autóbuszát. A villamosnak és az egyik autóbusznak már két-két szervizelését igazoló munkalapja is legyen, amelyek közül az egyik még befejezetlen (vég-dátuma üres). Látszódjék továbbá három cég, amely a szervizeléseket végezte vagy végzi, közülük az egyik két különböző jármű szervizeléséért is feleljen.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy jármű állapotgépét! Különböztesse meg a „szolgáltatban” és a „szervízben” van állapotait. Tervezze meg az állapot-átmeneteket megvalósító tevékenységeket, amelyeket majd az jármű osztály metódusaiként definiálhat. Ezek a metódusok például a szervizelést végző cég metódusait hívhatják.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni a modell alapján egy önkormányzat járműparkját, szervizelés végző cégeket, és a járművek szervizeléseit! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

2. Modellezzünk egy **autókereskedés** vásárlásait és eladásait!

Az autókereskedések különféle partnerekkel kereskednek: autót vesznek, eladnak. Egy szerződés tartalmazza, hogy melyik autókereskedés, melyik partnerrel, melyik autóra, mikor kötötte, és mennyiért, továbbá, hogy az autókereskedés szempontjából ez egy vételi, vagy eladási szerződés-e. A szerződésben feltüntetett ár különbözhet az autó értékétől. Egy autónak ismert az azonosítója (sztring), márkája (például Audi, Mazda, Skoda), gyártási éve (gyév), hajtó anyaga (benzin, dízel, elektromos, hibrid), újkori ára (újár). A jármű aktuális értéke $újár \cdot (100 - (jelen - gyév)) / (100.0 \cdot faktor)$ (egészre kerekítve), ahol a faktor az autó korától és márkájától függő pozitív valós szám.

Faktor	benzin	dízel	elektromos	hibrid
Audi	1.0	0.9	1.2	1.3
Mazda	2.0	2.0	2.5	2.3
Skoda	3.0	3.1	3.8	4.0

- Mennyi egy adott kereskedésben árult Audik értéke összesen?
- Van-e egy adott kereskedésben adott évnél fiatalabb Skoda?
- Mekkora a nyeresége egy adott kereskedésnek: az eladási szerződéseken szereplő árak összegéből vonjuk le a vételi szerződésen szereplő árak összegét?
- Hány szerződést kötött egy adott autókereskedés egy adott partnerrel?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutasson egy kereskedést, két autót, az egyik legyen jelenleg is a kereskedés tulajdona (eggyel több vételi szerződés van rá, mint eladási), a szerződésekhez kapcsolódjanak a szerződő partnerek. Ábrázoljuk azt, hogy az egyik autót, amelyet a kereskedés az egyik partnertől megvásárolt, annak a másik partnernek adta el, akitől a másik autót vásárolta. Ezt a másik autót a kereskedés egy harmadik partnernek eladta, majd visszavásárolta.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy autó állapotgépet, abból a szempontból, hogy egy adott autókereskedéshez tartozik-e vagy sem. Különböztesse meg az ott „árulják”, illetve a „nem tulajdon” állapotokat. Az állapotátmeneteket megvalósító tevékenységeket majd az autó osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni a modell alapján egy autókereskedéssel kapcsolatban álló autókat, partnereket, és az adás-vételeket. Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

3. Modellezzük egy **kisállatkereskedés** tevékenységeit!

Egy állatkereskedésben különféle állatokat (hörcsög, pinty, tarantulla, ...) árulnak. Egy állatot egyedi sztring azonosít, ismert a színe, az értéke, de az aktuális árát úgy számoljuk ki, hogy az értékét az állat jellemzőjétől (pl. fiatal vagy felnőtt) függő szorzótényezővel korrigáljuk. Az állatkereskedések nyilvántartják a partnereiket: akiktől beszerezik az állatokat, illetve akiknek eladják azokat. Mind a beszerzéseket, mind az eladásokat számlák igazolják. Egy számla (ami tehát lehet beszerzési vagy eladási) tartalmazza, hogy melyik kereskedés melyik partnerrel melyik kisállatra mikor és mennyiért kötött üzletet. Ugyanazon állat ára szerződésenként eltérő lehet, de ezek alapján kiszámolható egy átlagár.

Szorzó	fiatal	felnőtt
hörcsög	2.0	1.0
pinty	1.0	3.0
tarantulla	3.0	2.0

- Van-e egy kereskedésben adott színű pinty?
- Hány hörcsöge van egy kereskedésnek?
- Melyik a legnagyobb eszmei értékű tarantullája egy kereskedésnek?
- Hány szerződést kötött egy adott kereskedés egy adott partnerrel?
- Mekkora egy kereskedésnek a nyeresége (eladási számláin szereplő árak összege mínusz a beszerzési számláin szereplő árak összege)?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat egy kereskedést, három partnert, két állatot, és néhány szerződést; az állatok közül az egyik legyen éppen a kereskedés tulajdona (eggyel több vételi szerződése van rá, mint eladási).

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy kereskedés állapotgépét! Egy kereskedés állapotát a kereskedésben árult állatok gyűjteménye határozza meg: ennek megfelelően a kereskedés lehet üres vagy nem üres. Az állapotátmeneteket megvalósító tevékenységeket majd a kereskedés osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég "..."-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy kereskedésben árult állatokat, a kereskedés partnereit, és az állatok adás-vételeit. Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

4. Modellezzük egy **kamionos** cég működését!

A cég telephelyekkel (ezeket a címük azonosítja), kamionokkal és sofőrökkel rendelkezik. Egy kamionoknak ismert a rendszáma, a tengelyeire vetített terhelhetősége, fogyasztása (100km-en), tengelyeinek száma (nyergesvontató: 3, fülkésváz: 2). A kamionok vagy úton vannak, vagy az egyik telephelyen állnak.

- a. Melyik telephelyhez tartozik éppen most a legkevesebb kamion?
- b. Mekkora terhet képesek egyszerre elfuvarozni egy adott telephely kamionjai?

Amikor egy kamionok egy fuvarra kap megbízást, akkor meg kell adni a fuvar távolságát, súlyát, díját (amit a cég kap a szállításért), az indulási dátumát, tervezet szállítási időt (órában), és a sofőrt. A fuvar teljesítése után ismert lesz annak érkezési ideje (addig ez egy extrémális érték). Új megbízást nemcsak egy telephelyen állomásozó kamion kaphat, hanem egy már úton levő is, ha annak nincs teljesítendő fuvarja.

- c. Volt-e olyan kamion, amely túl volt terhelve valamelyik fuvarban?
- d. Mekkora a nyeresége a cégnek? (fuvar nyeresége = szállítási díj – üzemanyag költség – bér)?

A sofőrök között vannak kezdők, több éves gyakorlattal rendelkezők, illetve a cég törzsgárdájához tartozók. A sofőrök a teljesített fuvarjaik után kapnak bért, amelynek összege a fuvarral megtett távolságtól, a kamion típusától (nyerges, fülkés), illetve a sofőr besorolásától (kezdő, gyakorlott, törzstag) függ az alábbi módon: $\text{bér} = \text{távolság} \cdot \text{együttható}$

együttható	kezdő	gyakorlott	törzstag
nyerges	25	35	40
fülkés	20	30	40

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat két telephelyet, két sofőrt, három kamiont, és néhány fuvart.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy kamion objektum állapotgépét! Különböztesse meg a „telephelyen áll”, „fuvar teljesít”, „üresen közlekedik” (el a telephelyről, vagy vissza a telephelyre) állapotokat. Az állapot-átmeneteket megvalósító tevékenységeket majd az vontató osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy cég telephelyeit, vontatóit, sofőrjeit, fuvarjait. Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

5. Modellezzük egy **könyvtár** működését!

A könyvtár nyilvántartja a könyvtárba beiratkozott személyeket, és a kikölcsönözhető könyveit. Könyvtári tag az a személy lehet (ismerjük a nevét), aki beiratkozik a könyvtárba.

Egy könyvtári tag egy alkalommal legfeljebb öt, a könyvtárban meglévő könyvet kölcsönözhet ki. Egy könyvnek ismert a címe, szerzője, kiadója, ISBN száma, az oldalszáma, és van egy könyvtári azonosítója, miután a könyvtárba kerül. Az egyszerre kikölcsönzött könyveket több részletben is vissza lehet hozni, így egy kölcsönzés eseményhez tartozó könyvek listája folyamatosan csökkenhet.

Egy könyv kölcsönzési pótdíja a kölcsönzés lejáratí idejétől számított napok számától függ, de az egy napi pótdíj a könyv műfajától (természettudományi, szépirodalmi, ifjúsági) és példányszámát jellemző kategóriától (ritkaság, sok példány) függ.

napi pótdíj	ritkaság	sok példány
természettudományi	100	20
szépirodalmi	50	10
ifjúsági	30	10

- Tegye lehetővé, hogy a könyvtár beszerezhesen egy könyvet, egy új személy be tudjon iratkozni, egy tag kikölcsönözhesse az általa kért könyvek közül azokat, amelyek jelenleg elérhetők, és bármikor visszahozhasson egy kikölcsönzött könyvet.
- Mennyi pótdíjat kell fizetnie egy tagnak a vissza nem hozott könyvei után?

Készítsen használati eset diagramot!Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat öt könyvet, két könyvtári tagot, hozzájuk kapcsolható három kölcsönzési tevékenységet, ahol az egyik kölcsönzés egyszerre két könyvet is tartalmaz.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy könyv objektum állapotgépét! Különböztesse meg a „könyvtárban”, és a „kikölcsönözve” állapotokat. Az állapot-átmeneteket megvalósító tevékenységeket majd a könyv osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég ”...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy könyvtár könyveit, könyvtári tagjait, néhány kölcsönzést és könyv visszahozást. Számoljuk ki egy tag pótdíját! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

6. Modellezzük egy **teniszkлуб** működését!

A klub nyilvántartja a klubtagjait, akik foglalást tehetnek a klub szabad tenispályáira. A teniszkлуб pályái lehetnek füvesek, salakosak, vagy műanyag. A pályákat egy sorszám azonosítja. A füves pálya óradíja 5000 Ft, a salakosé 3000 Ft, a műanyagé 2000 Ft. A pályák egy része még sátorral is fedett: ezen pályák óradíjához 20%-os felárat kell fizetni. Minden foglalás 1 órára szól, amely tartalmazza a foglaló klubtag nevét, a választott pálya sorszámát, a foglalás dátumát, és a lefoglalt órát (6-20 közötti szám).

Tegye lehetővé, hogy a klub új pályát tudjon létrehozni, egy régit fel tudjon számolni, egy személy be-, illetve kiléphessen a klubból, egy tag időpontot tudjon foglalni egy pályára, vagy akár vissza is mond hasson egy foglalást. Meg lehessen válaszolni az alábbi kérdéseket:

- a. Mennyi pályahasználati díjat kell fizetnie az adott napra egy adott tagnak?
- b. Mennyi a teniszkлуб aznapi összbevétele?
- c. Keressünk egy adott időpontra megadott borítású foglalható pályákat.
- d. Mondjuk meg mely pályákat foglalta le egy tag egy adott napra és mikor?

Készítsen használati eset diagramot a klub és egy klubtag szempontjából! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat öt pályát, két teniszkлубtagot, ezekhez kapcsolódó 2-2 pályafoglalást. Egy kommunikációs diagramban jelölje, hogy mely objektumok milyen metódusokkal kell, hogy rendelkezzenek ahhoz, hogy a kívánt funkcionalitást biztosítani tudjuk.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég "..."-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy teniszkлуб pályáit, tagjait, a tagok pályafoglalásait, illetve tartalmazza ezek megszüntetését. Adjuk meg a választ a feladatban feltett kérdésekre! Készítsen teszteteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

7. Modellezzük egy **filmszínház** jegyvásárlásait!

A filmszínház több vetítőteremben vetít a filmeket megadott időintervallumokban (előadások). A vetítőterem eltérő méretűek: széksoraikat betűk (A, B, C, ...), a sorok ülőhelyeit számok (1, 2, 3, ...) azonosítják. A nézők előre foglalhatnak helyet egy előadásra (terem és időintervallum), majd a jegyet megvásárolják. Foglалás nélkül is lehet szabad helyet vásárolni. Egy jegy árara (természetes szám) kedvezmény adható attól függően, hogy mi vetítőterem kategóriája (nagy, közepes, VIP) illetve, hogy a birtokosa (tehát nem az, aki megvette) felnőtt, diák, gyerek, nyugdíjas vagy törzstag: $\text{jegyár} = \text{alapár} \cdot ((100 - \text{kedvezmény})/100.0)$

kedvezmény	gyerek	diák	felnőtt	nyugdíjas	törzstag
kis terem	40	30	10	30	30
nagy terem	40	20	0	20	30
VIP terem	0	0	0	0	0

- Melyik filmet nézte meg a legtöbb néző?
- Számoljuk meg egy adott előadásra megvett, csak lefoglalt, illetve szabad helyeket!

Készítsen használati eset diagramot! Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely megmutatja egy filmszínház egy vetítőtermét, ahhoz rendelt két előadást, azt, hogy egyik előadásra két helyet már egy néző lefoglalt, egy harmadikat, egy gyerek számára valaki már kifizetett.

Rajzolja fel a feladat osztály diagramját! A triviális getter/setter-eket ne írja bele a modellbe: azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölje publikusnak; a csak getter-rel rendelkező adattag mellé írjon „readonly”-t.

Készítse el egy előadás objektum állapotgépét! Ennek egy állapotát az ülőhelyeinek állapotai együttesen határozzák meg. Egy ülőhely állapota lehet „szabad”, „foglalt”, vagy „eladott”. Tervezze meg az állapot-átmeneteket megvalósító tevékenységeket, amelyeket majd az Előadás osztály metódusaiként definiálhat.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni a modell alapján egy filmszínház terveit, előadásait, foglalásait és jegyvásárlásait! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

További ötletek:

8. A **Jóétvágyat Kft.** a Balaton strandjain büféket üzemeltet. A büfék egyedi azonosítóval rendelkeznek és ismert a helyszínük. A büfékben sült csirkecomb (250 Ft/10 dkg), rántott húst (300 Ft/10 dkg), illetve hekket (400 Ft/10 dkg), lehet kapni. Mindegyiket rendelhetjük önmagában, vagy kis, közepes illetve nagy menüben is, azaz sült krumplival és üdítővel. Az egyes tételek árait a súlyuk (dkg) és egységáruk szorzatával számoljuk ki, melyhez hozzáadjuk a menü kiegészítő árát (kicsi: 550 Ft, közepes: 700 Ft, nagy: 900 Ft)

A vendégek vásárlói kártyát kapnak a Kft-től, amely egyedi kártyaszámmal rendelkezik. A kártyájuk alapján ismert a vendégekről, hogy az egyes büfékben milyen tételt vásároltak.

- Tegye lehetővé, hogy egy vásárló kártyát kapjon a Kft-től, majd vásárolhasson annak büféiben.
- Igaz-e, hogy a Kft összes büféjének volt olyan vendége, aki 10.000 forintnál többet költött?
- Mennyit költött egy vendég egy adott büfében?
- Hány vendég költött 10.000 forintnál többet egy adott büfében?

Készítsen használati eset diagramot! Ebben jelenjenek meg használati esetként a később bevezetett fontosabb metódusok. Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat két büfét, három vendéget, és néhány vásárlást.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy vendég objektum állapotgépét! Különböztesse meg a „nincs vásárló kártyája”, „van vásárló kártyája” állapotokat. Az állapot-átmeneteket megvalósító tevékenységeket majd az vendég osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik a paraméterek alapján, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modellt! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni a Kft. büféit, a vendégeket, és a vásárlásaikat. Készítsen tesztteseteket, néhányat rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

9. A **fagyizók** nevét és címét ismerjük, nevük egyedi. A fagyizókban háromféle fagylalt-kehely rendelhető: kicsi, közepes és nagy. A fagylalt-kelyhek három vagy öt gombócból állnak, a gombóc súlya kicsi esetén 55g, közepes esetén 70g, a nagy pedig 90g.

A fagyizóban nyilvántartják az aznapi kínálatot: milyen ízű fagyiból lehet választani, valamint az aznapi rendeléseket. A cég fagyizói egymástól eltérő áron adhatják a fagyit, amelynek ára annak ízétől független, és egy napon belül nem változik. Egy megrendelést az időpontja azonosít.

- a. Tegye lehetővé, hogy megrendeléseket lehessen feladni egy fagyizóban.
- b. Hány epret is tartalmazó fagylalt-kelyhet adtak el egy fagyizóban egy adott méretű kehelyből?

Mennyi egy fagyizó napi bevétele?

Készítsen használati eset diagramot! Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely mutat két telephelyet, öt sofőrt, három vontatót, és néhány fuvart.

Rajzolja fel a feladat osztály diagramját (először csak a konstruktorokkal)! Azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölheti publikusnak. (A triviális getter/setter-eket később sem kell beírni a modellbe.)

Készítse el egy vontató objektum állapotgépét! Különböztesse meg a „telephelyen áll”, „fuvart teljesít”, „üresen közlekedik” (el a telephelyről, vagy vissza a telephelyre) állapotokat. Az állapot-átmeneteket megvalósító tevékenységeket majd az vontató osztály metódusaiként definiálhatja.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik a paraméterek alapján, nem kell feltüntetni. Ilyenkor a konstruktor paraméterlistája helyén elég felsorolni az inicializálandó adattagok neveit. Az összes közvetlen (tehát nem szerepnév) adattag felsorolása helyett elég „...”-ot írni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modelljét! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni egy cég telephelyeit, vontatóit, sofőrjeit, fuvarjait. Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

10. A **Tengerimalac** Tenyésztők Egyesülete kiállításokat szervez azokkal a tenyésztőkkel, akik tagjai a szervezetnek. Egy kiállításokon több tenyésztő több tengerimalaccal is indulhat. Ugyan az a tenyésztő több kiállításon is részt vehet. A kiállításoknak ismert a helyszíne és a dátuma. A tenyésztőknek ismert a neve.

A tengerimalacoknak van tenyésztője, adott a fajtája (rövidszőrű: angol rövidszőrű (azaz Arsz), Rex, Teddy; hosszúszőrű: Texel, Shelti, Perui; félhosszú szőrű: Abesszin- és Svájci Teddy. A tengerimalacokat kondíció, habitus, fej, fül, fajtaspecifikus tulajdonságaik alapján pontozzák. (Ez utóbbi az Arsz, Rex, Teddy esetén a szín egyenletesség, Texel, Shelti, Perui esetén a szőrhossz, Abesszin- és Svájci Teddy esetén a forgók száma.) Minden malac minden kategóriában max. 20 pontot kaphat, és ezeket a pontokat a kategóriától és a malac fajtájától függő szorzóval számítják be a végeredménybe.

- a. Tegye lehetővé, hogy egy tenyésztő tagja lehessen az egyesületnek, majd regisztrálhasson egy kiállításra, ahol az általa kiállított állatait értékeljék.
- b. Melyik tenyésztő malaca szerezte a legtöbb pontot egy adott kiállításon?
- c. Van-e olyan tenyésztő, akinek minden kiállításon volt olyan malaca, ami a maximális pontszámot kapta?

szőrzet	kondíció	habitus	fej	fül	fajta
rövid	40%	10%	10%	10%	30%
hosszú	20%	10%	10%	10%	50%
félhosszú	15%	15%	5%	5%	60%

11. A **turisták** látogatása bevételt hoz egy városnak, de kis mértékben rontja is a város állapotát. Egy város, ami jó állapotban van, vonzza a turistákat, míg a rossz állapotú város taszítja az odalátogatni készülőket.

Egy turista látogatása átlagosan 100.000 Ft bevételt hoz a városnak. Ha a város bevétele egy évben meghaladja az 1 milliárd forintot, akkor az 1 milliárdon felüli részt a város javítására és szépítésére fordítja. Egy város állapotát 1-100-ig terjedő skálán értékeljük (1 alá és 100 fölé sose megy, mert az állam elkölti a fölösleget, de besegít, ha már nagyon vészes a helyzet). Ezen belül 1 és 33 közt számít lepusztultnak, 34 és 67 között átlagosnak és 67 fölött jó állapotúnak. Húszmillió forint hoz egy pont állapotjavulást a városnak.

A turistáknak 3 fajtája van: japánok, akik nem rontanak a város állapotán (rendet raknak maguk után); a nyugati országokból érkező turisták, akik minden 100 fő esetén újabb egy pontot rontanak a város állapotán (kevésbé ügyelnek a környezetükre), és a többiek, akik már minden 50 fő esetén újabb egy pontot rontanak a város állapotán (a szemetelés kulturális szokásnak tekinthető).

Ha egy év elején a város jó állapotban van, akkor 20%-kal több japánt és 30%-kal több nyugatit vonz, mint ahány tervezte, hogy ellátogat oda. Átlagos állapotban 10%-kal több nyugatit és 10%-kal több egyéb típusú turistát vonz. Lepusztult állapot esetén a japánok egyáltalán nem jönnek, a többiek pedig csak annyian, amennyien tervezték.

A program egy szövegfájlból olvassa be az adatokat! Az első sorban a város kezdeti állapota szerepel. A második sor jelöli a szimulált évek számát. A következő sorok tartalmazzák, hogy az egyes években hány turista tervezte, hogy eljön a városba: minden sor 3 darabszámot tartalmaz (japánok, modernnek, többiek). A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

50
3
1000 4000 6000
2000 3000 8000
6500 5000 3000

Adjuk meg, hogy a fájlban jelölt évek letelte után milyen a város állapota! Körönként mutassuk meg az érkezett turisták számát (hány tervezett és hány jött) kategóriák szerint, az éves bevételt és a város felújítás előtti állapotát (szám és kategória)!

12. A **tundra élővilágát** zsákmányállat-kolóniák és ragadozókolóniák alkotják, amelyek egyedszámai hatással vannak a többire. Ha a zsákmányállatok száma nagyon nagy, nagyrésztük elvándorol, mert nem talál élelmet. Ha a ragadozók száma nagy, gyorsabban fogy a zsákmányállatok száma, mert vadásznak rájuk.

Három ragadozófajt különböztetünk most meg: hóbaglyot, sarki rókát és farkast. Háromfajta zsákmányállat van: lemming, sarki nyúl és ürge. Az egyes kolóniáknak van beceneve, faja, illetve egyedszáma. A következőkben megadjuk, hogy a zsákmányállat-kolóniák miként reagálnak a rájuk vadászó ragadozókolóniákra:

Lemming: ha vadásznak rájuk, az egyedszámuk a rájuk vadászó kolónia számának 4-szeresével csökken. Minden második körben megduplázódik az egyedszámuk. 200 fölötti egyedszámnál csökken a számuk 30-ra.

Nyúl: ha vadásznak rájuk, az egyed számuk a rájuk vadászó kolónia számának 2-szeresével csökken. Minden második körben másfélszeresére nő az egyedszámuk. 100 fölötti egyedszámnál csökken a számuk 20-ra.

Ürge: ha vadásznak rájuk, az egyed számuk a rájuk vadászó kolónia számának 2-szeresével csökken. Minden negyedik körben megduplázódik az egyedszámuk. 200 fölötti egyedszámnál csökken a számuk 40-re.

A ragadozó kolóniák minden körben véletlenszerűen kiválasztanak egy zsákmány kolóniát, amit megtámadnak. Ha nincs elegendő állat a megtámadott kolóniában (azaz nincs annyi egyed, amennyivel fogy a zsákmányok száma, ha megtámadják őket), akkor fogy a ragadozók száma is: közül minden negyedik elpusztul. Minden nyolcadik körben a hóbaglyok 4 egyedenként 1, a rókák 4 egyedenként 3, a farkasok 4 egyedenként 2 utódot fialnak.

Addig szimuláljuk a folyamatot, amíg minden ragadozó kolónia egyedszáma 4 alá nem csökken, vagy a ragadozók összesített száma meg nem duplázódik a kiinduló értékhez képest. Körönként mutassuk meg a kolóniák összes tulajdonságát!

A program egy szövegfájlból olvassa be a kolóniák adatait! Az első sorban a zsákmánykolóniák és a ragadozó kolóniák darabszámai szerepelnek szóközzel elválasztva. A következő sorok tartalmazzák a kolóniák adatait szóközzel elválasztva: a becenevüket (szóközők nélküli sztring), a fajukat (amit egy karakter azonosít: h - hóbagoly, s – sarki róka, f – farkas, l – lemming, n – nyúl, u - ürge), és a kezdeti egyedszámukat.

A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4 2
kicsik l 86
picik l 90
szaporak n 26
szorgosak u 12
ehesek f 12
tollasak h 6

13. Egy **királyság** földjeit ellenség szállta meg. A királyság felhívást intézett a környező vidékeken, kalandozókat hívott, akik megtisztítják a földeket az ellenséges csapatoktól. A felhívásra több kalandozó is jelentkezik, akik egy csapatban indulnak neki, hogy sorban végigjárva a földeket, elűzzenek onnan minden ellenséget. A kalandozók három csoportba tehetők: vannak az óvatosak, a magabiztosak és a bátrak. Az óvatosak kevesebb sérüléssel is visszatérnek a várba gyógyulni, így majdnem minden küldetést túlélnek. A magabiztosak kicsit realitásabban saccolják meg, hogy mennyit élnek túl, így ők tovább terepen maradnak. A bátrak addig mennek előre, amíg még éppen, hogy bírják. A földek lehetnek az ellenség kezén, ekkor harcolni kell velük. Lehetnek senkié, amikor a lakosok elmenekültek az ellenség elől, mielőtt odaért volna. Ezenfelül tartozhatnak baráthoz is, ami lehetővé teszi a kalandozók megerősödését. Amint a kalandozók végigjárnak egy üres földet, a lakosok visszatérnek oda, mert biztonságosnak érzik. Egy ellenségtől megtisztított terület addig nem tartozik senkihez, amíg a kalandozók újra végig nem járják. A kalandozóknak három lehetőségük van végigjárni minden területet. Ha ezalatt nem sikerül megtisztítani a földeket, az ellenség nagy erővel támadásba lendül és a küldetés elvész. Ha sikerül három próbálkozással végigjárni és megtisztítani minden földet, a küldetés sikeres.

A kalandozók fejenként legalább 1, legfeljebb pedig a típusuktól függő életerőt veszíthetnek ellenséges földeken. A tényleges sebződést egy 1 és (n / a) mozgásban lévő kalandozók száma) közti véletlenszám adja meg mindegyikükénél, ahol n a bátrak esetében 100-at, a magabiztosak esetében 90-et, az óvatosak esetében 80-at jelent. Barát földjén ugyanezen szabály szerint gyógyulnak (a bátor merészebben halad előre az ellenséges területen és bátrabban kér segítséget is a barátoktól).

A kalandozók, ha már nagyon kicsinek érzik az életerejüket, visszatérnek a várba, ahol meggyógyulnak teljesen. A küldetésbe csak akkor kapcsolódnak be újra, ha nincs több mozgásban lévő kalandozó (meghalnak vagy ők is visszatérnek). A kalandozók a következő életerők alatt térnek vissza: óvatos – 41, magabiztos – 21, bátor – 11.

Egy kalandozó maximális életeroje 100. Ha elfogy az életeroje ellenséges területen, meghal.

Az adatok beolvasása fájlból történik. A fájl első sora tartalmazza a kalandozók számát, majd minden sorban egy-egy kalandozó adatai találhatók: a típusa, a neve és az életeroje. Az utolsó sorban a földek típusa van jelölve egy karaktorsorozattal. Pl. a 'EFEN' sorozat azt jelenti, hogy a királyság 4 földdel rendelkezik, egy ellenséges, egy barát, egy ellenséges és egy senkiföldje területtel.

kedvezmény	előétel	főétel	desszert	üdítő ital
normál vendég	0	0	0	0
törzsvendég	0	10	5	0
nagy csoport	5	15	10	10

- Tegyük lehetővé az asztalfoglalásokat! Rögzítsük, amikor egy vendég leül egy helyre!
- Listázzuk ki egy adott időszakra a megadott számú hellyel rendelkező szabad asztalokat!
- Tegyük lehetővé a vendégek rendeléseinek rögzítését!
- Számoljuk ki azt a végösszeget, amelyet egy vendégnek fizetnie kell az általa elvállalt rendelések alapján!
- Összegezzük egy nap teljes bevételét!

Körönként írja ki, hogy milyen területen haladnak át és mennyi az életerejük (ha meghalnak, vagy visszatérnek a várba, azt is). Adja meg, hogy a küldetés sikeres-e! A küldetés végén méltó tisztelettel emlékezzen meg a hősi halottakról!

Készítsen használati eset diagramot! Adjon meg a fenti feladathoz egy olyan objektum diagramot, amely megmutatja egy vendéglő két asztalát: az egyikén legyen két foglalás, a másikon üljön két vendég, akiknek már vannak rendelései.

Rajzolja fel a feladat osztály diagramját! A triviális getter/setter-eket ne írja bele a modellbe: azoknak a privát/védett adattagoknak a láthatóságát, amelyekhez getter-t is, és setter-t is kell készíteni, jelölje publikusnak; a csak getter-rel rendelkező adattag mellé írjon „readonly”-t.

Készítse el egy ülőhely objektum állapotgépét! Különböztesse meg annak „szabad”, „foglalt”, és „fenn tartott” állapotait. Tervezze meg az állapot-átmeneteket megvalósító tevékenységeket, amelyeket majd az ülőhely osztály metódusaiként definiálhat.

Egészítse ki az osztálydiagramot az objektum-kapcsolatokat létrehozó metódusokkal, valamint a feladat kérdéseit megválaszoló metódusokkal. A metódusok leírásában a félév első felében bevezetett végrehajtható specifikációs jelöléseket használja. Azoknak a konstruktoroknak a törzsét, amelyek kizárólag az adattagok inicializálását végzik a paraméterek alapján, nem kell feltüntetni.

Használjon tervezési mintákat, és mutasson rá, hogy hol melyiket alkalmazta.

Implementálja a modelljét! Szerkesszen olyan szöveges állományt, amelyből fel lehet populálni a modell alapján egy vendéglő asztalait, majd asztalfoglalásokat, rendeléseket, fizetéseket lehet szimulálni! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

14. Egy **bolygón** különböző fajtájú növények élnek, minden növény tápanyagot használ. Ha egy növény tápanyaga elfogy (a mennyisége 0 lesz), a növény elpusztul. A bolygón háromféle sugárzást különböztetünk meg: alfa sugárzás, delta sugárzás, nincs sugárzás. A sugárzásra a különböző fajtájú élő növények eltérő módon reagálnak. A reakció tartalmazza a tápanyag változását, illetve a következő napi sugárzás befolyásolását. A másnapi sugárzás alakulása: ha az alfa sugárzásra beérkezett igények összege legalább hárommal meghaladja a delta sugárzás igényeinek összegét, akkor alfa sugárzás lesz; ha a delta sugárzásra igaz ugyanez, akkor delta sugárzás lesz; ha a két igény közti eltérés háromnál kisebb, akkor nincs sugárzás. Az első nap sugárzás nélküli.

Minden növény jellemzői: az egyedi neve (sztring), a rendelkezésre álló tápanyag mennyisége (egész), hogy él-e (logikai). A szimulációban részt vevő növények fajtái a következők: puffancs, deltafa, parabokor. A következőkben megadjuk, hogy az egyes fajták miként reagálnak a különböző sugárzásokra. Először a tápanyag változik, és ha a növény ezután él, akkor befolyásolhatja a sugárzást.

Puffancs: Alfa sugárzás hatására a tápanyag mennyisége kettővel nő, sugárzás mentes napon a tápanyag eggyel csökken, delta sugárzás esetén a tápanyag kettővel csökken. Minden esetben úgy befolyásolja a másnapi sugárzást, hogy az 10 egységgel növeli az alfa sugárzás igényét. Ez a fajta akkor is elpusztul, ha a tápanyag mennyisége 10 fölé emelkedik.

Deltafa: Alfa sugárzás hatására a tápanyag mennyisége hárommal csökken, sugárzás nélküli napon a tápanyag eggyel csökken, delta sugárzás hatására a tápanyag négyvel nő. Ha a tápanyag mennyisége 5-nél kisebb, akkor 4 egységgel növeli a delta sugárzás igényét, ha 5 és 10 közé esik, akkor 1 értékben növeli a delta sugárzás igényét, ha 10-nél több, akkor nem befolyásolja a másnapi sugárzást.

Parabokor: Akár alfa, akár delta sugárzás hatására a tápanyag mennyisége eggyel nő. Sugárzás nélküli napon a tápanyag eggyel csökken. A másnapi sugárzást nem befolyásolja.

Szimuláljuk a növények viselkedését és adjuk meg, hogy x nap után melyik életben maradt egyed a legerősebb! Minden lépésben írjuk ki az összes növényt a rájuk jellemző tulajdonságokkal, valamint az aktuális sugárzást!

A program egy szövegfájlból olvassa be a szimuláció adatait! Az első sorban a növények száma szerepel. A következő sorok tartalmazzák a növények adatait szóközzel elválasztva: a növény nevét, a fajtáját és a kezdetben rendelkezésére álló tápanyag mennyiségét. A fajtát egy karakter azonosítja: p - puffancs, d - deltafa, b - parabokor. A növényeket leíró részt követő sorban a szimuláció napjainak száma adott egész számként. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4 Falánk p 7 Sudár d 5 Köpcös b 4 Nyúlánk d 3 10

15. Az **időjárás** a levegő aznapi páratartalmától függ: Ha ez meghaladja a 70%-ot, esős idő lesz, és ekkor lecsökken a páratartalom 30%-ra. 40%-os páratartalom alatt az időjárás napos lesz. 40 és 70% közötti páratartalom esetén az esős időjárásnak $(\text{páratartalom}-40)*3,3$ százalék az esélye, egyébként felhős időjárás lesz. (Véletlenszám generátorral állítsunk elő egy számot 0 és 100 között, és ha ez kisebb, mint $(\text{páratartalom}-40)*3,3$ érték, akkor esős, különben felhős időjárás legyen.)

Az egyes földterületek – a megadásuk sorrendjében – reagálnak a különböző időjárásokra: először a vízmennyiségük változik, majd befolyásolják a levegő páratartalmát. Egyetlen földterület vízmennyisége sem lehet negatív.

Pusztá: napos idő hatására a vízmennyiség 3 km^3 -rel csökken, felhős idő hatására 1 km^3 -rel nő, eső hatására 5 km^3 -rel nő. A levegő páratartalmát 3%-kal növeli. 15 km^3 -nél több tárolt víz esetén zölddé változik.

Zöld: napos idő hatására a vízmennyiség 6 km^3 -rel csökken, felhős idő hatására 2 km^3 -rel, eső hatására 10 km^3 -rel nő. A levegő páratartalmát 7%-kal növeli. 50 km^3 -es vízmennyiség fölött tavassá változik. 16 km^3 alatt pusztává változik.

Tavas: napos idő hatására a vízmennyiség 10 km^3 -rel csökken, felhős idő hatására 3 km^3 -rel, eső hatására 15 km^3 -rel nő. A levegő páratartalmát 10%-kal növeli. 51 km^3 alatt zölddé változik.

Adjuk meg 10 kör után a legvizesebb földterület tulajdonosát, vízmennyiségével együtt! Körönként mutassuk meg a földterületek összes tulajdonságát!

A program egy szövegfájlból olvassa be az adatokat! Ennek első sorában a földterületek száma szerepel. A következő sorok tartalmazzák a földterületek adatait szóközzel elválasztva: a terület tulajdonosát (szóköz nélküli sztring), fajtáját (egy karakter azonosítja: p - pusztá, z - zöld, t - tavas), és a kezdeti vízmennyiségét. Az utolsó sor a földterületek feletti levegő kezdeti páratartalmát mutatja. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4 Bean t 86 Green z 26 Dean p 12 Teen z 35 98
--

16. A Föld **hidrológiai körfolyamatában** a különböző földterületek befolyásolják az időjárást és a különböző időjárások hatására a földterületek változnak. Minden földterületnek van neve, fajtája (puszta, zöld, tavas), tárolt vízmennyisége (km^3 -ben). A földterületek feletti közös levegőnek ismerjük a páratartalmát (százalékban).

Az időjárás a levegő aznapi páratartalmától függ: Ha ez meghaladja a 70%-ot, esős idő lesz, és ekkor lecsökken a páratartalom 30%-ra. 40%-os páratartalom alatt az időjárás napos lesz. 40 és 70% közötti páratartalom esetén az esős időjárásnak $(\text{páratartalom}-40)*3,3$ százalék az esélye, egyébként felhős időjárás lesz. (Véletlenszám generátorral állítsunk el egy számot 0 és 100 között, és ha ez kisebb, mint $(\text{páratartalom}-40)*3,3$ érték, akkor esős, különben felhős időjárás legyen.)

Az egyes földterületek – a megadásuk sorrendjében – reagálnak a különböző időjárásokra: először a vízmennyiségük változik, majd befolyásolják a levegő páratartalmát. Egyetlen földterület vízmennyisége sem lehet negatív.

Puszta: napos idő hatására a vízmennyiség 3 km^3 -rel csökken, felhős idő hatására 1 km^3 -rel, eső hatására 5 km^3 -rel nő. A levegő páratartalmát 3%-kal növeli. 15 km^3 -nél több tárolt víz esetén zölddé változik.

Zöld: napos idő hatására a vízmennyiség 6 km^3 -rel csökken, felhős idő hatására 2 km^3 -rel, eső hatására 10 km^3 -rel nő. A levegő páratartalmát 7%-kal növeli. 50 km^3 -es vízmennyiség fölött tavassá változik. 16 km^3 alatt pusztává változik.

Tavas: napos idő hatására a vízmennyiség 10 km^3 -rel csökken, felhős idő hatására 3 km^3 -rel, eső hatására 15 km^3 -rel nő. A levegő páratartalmát 10%-kal növeli. 51 km^3 alatt zölddé változik.

Addig szimuláljuk a folyamatot újra és újra a földterületek megadott sorrendjében, amíg minden földterület azonos fajtájú nem lesz. Körönként mutassuk meg a földterületek összes tulajdonságát!

A program egy szövegfájlból olvassa be az adatokat! Ennek első sorában a földterületek száma szerepel. A következő sorok tartalmazzák a földterületek adatait szóközzel elválasztva: a terület tulajdonosát (szóköz nélküli sztring), fajtáját (egy karakter azonosítja: p - puszta, z - zöld, t - tavas), és a kezdeti vízmennyiségét. Az utolsó sor a földterületek feletti levegő kezdeti páratartalmát mutatja. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4 Bean t 86 Green z 26 Dean p 12 Teen z 35 98
--

17. Ismerjük a **légkör** egymás felett elhelyezkedő ózon, oxigén, és széndioxid anyagú légrétegeit, amelyek vastagsága a légköri viszonyoktól (zivataros, napos, egyéb) függően változik. Amikor egy légköri réteg anyagának egy része átalakul, akkor ez az anyagmennyiség fölszáll, és vastagítja a felette lévő első ugyanolyan anyagú réteget. Ha nincs fölötte ilyen réteg, akkor a légkör legtetején új réteget képez. Egy rétegnek sem csökkenhet a vastagsága fél kilométer alá. Ha ez mégis megtörténne, akkor ez a réteg is felszáll, és egyesül a fölötte lévő első ugyanilyen anyagú réteggel. Ha azonban nincs ilyen, akkor megszűnik.

A folyamat során először egymástól függetlenül reagálnak az egyes légrétegek az aktuális időjárási viszonyra, utána rétegenként alulról felfelé haladva felszállnak az újonnan keletkeztek anyagmennyiségek, illetve a túl vékony rétegek.

A következőkben megadjuk, hogy az egyes anyagok miként reagálnak a különböző időjárási viszonyokra.

anyag	zivatar	napos	más
ózon	-	-	5% oxigénné
oxigén	50% ózonná	5% ózonná	10% széndioxiddá
széndioxid	-	5% oxigénné	-

Addig szimuláljuk a folyamatot, amíg el nem fogy valamelyik anyag teljesen a légkörből. Körönként mutassuk meg a légrétegek összes tulajdonságát!

A program egy szövegfájlból olvassa be a légkör adatait! Az első sorban a légrétegek száma szerepel. A következő sorok tartalmazzák alulról felfelé haladva a légrétegek adatait szóközzel elválasztva: anyaga (ezt egy karakter azonosítja: z - ózon, x - oxigén, s - széndioxid), és vastagsága. A rétegeket leíró részt követő sorban a változó légköri viszonyok találhatók egy karaktersorozatban (z - zivatar, n - napos, m – más). Ha a szimuláció a karaktersorozat végére ér, az elejéről folytatja.

A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

```
4
z 5
x 0.8
s 3
x 4
mmmmnnznmm
```

18. A **Föld megfigyelőállomásai** (obszervatóriumok, műholdak) rendszeresen kémlelik a világűrt, és feljegyzik az általuk észlelt űrbeli objektumok adatait: azonosítóját, típusát (aszteroida, fekete lyuk, vagy nap), a megfigyelés dátumát, a becsült tömegét, és a Földtől vett távolságát. A távolságot az aszteroidáknál ezer kilométerben (10^3 km), napoknál fényévben ($9,5 \cdot 10^{12}$ km), a fekete lyukaknál százmillió fényévben ($9,5 \cdot 10^{20}$ km) adják meg. A tömeget aszteroidáknál ezer tonnában (10^6 kg), fekete lyukak és napok esetén naptömegben ($1,98 \cdot 10^{30}$ kg) jegyzik fel. Egy objektum akár ugyanazon megfigyelőállomásnak több különböző megfigyelésében is szerepelhet. Természetes, hogy ugyanazon objektum különböző megfigyelései eltérő tömeg- és távolságbecslést tartalmaznak.

- a. Tegye lehetővé, hogy a Föld megfigyelőállomásokat hozhasson létre, majd ezek az állomások megfigyeléseket végezzenek különféle objektumokról, és ezeket rögzítsék.
- b. Hány állomáson figyeltek meg veszélyes objektumot? Egy objektum akkor veszélyes, ha az állomáson végzett megfigyelések szerint az átlagos tömege nagyobb, mint c_1 kg, és a Földhöz mért legközelebbi távolsága kisebb, mint c_2 km. (A c_1 és c_2 adott paraméterek.)
- c. Melyik állomáson figyelték meg a legnagyobb tömegű aszteroidát?
- d. A föld hidrológiai körfolyamatában a különböző földterületek befolyásolják az időjárást és a különböző időjárások hatására a földterületek változnak. Minden földterületnek van neve, fajtája (puszta, zöld, tavas), tárolt vízmennyisége (km^3 -ben). A földterületek feletti közös levegőnek ismerjük a páratartalmát (százalékban).

19. A **hobbi állatoknak** az életkedvük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a halaknak oxigén dús, megfelelő hőmérsékletű vízre; a madaraknak tágas, tiszta kalitkára; a kutyáknak rendszeres foglalkoztatásra. Pisti számos hobbi állatot tart: halakat, madarakat és kutyákat. Állatainak van neve és ismerhető az életkedvüket mutató 0 és 100 között szám (0 esetén az állat elpusztul). Pistinek vannak jobb és rosszabb napjai. Mikor nagyon jó kedvű, egyik állatáról sem feledkezik meg: ilyenkor a halak életkedve 1-gyel, a madaraké 2-vel, a kutyáké 3-mal nő. Átlagos napokon csak a kutyáival foglalkozik, a többi állat életkedve ilyenkor csökken: a halaké 3-mal, a madaraké 1-gyel. Amikor rosszkedvű, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit szomorúbb lesz: a halak 5 egységgel, a madarak 3-mal, a kutyák 10-zel.

Pisti kedve egy adott napon egy kategóriával jobb lesz attól, ha minden élő kedvencének az életkedve legalább 5.

Nevezze meg a legszomorúbb (legkisebb az életkedvű) állatot, amelyik még nem pusztult el a vizsgált napok után! Ha több ilyen életkedvű állat is létezik, akkor írja ki az összesnek a nevét! Naponként mutassa meg az állatok összes tulajdonságát!

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (H – hal, M – madár, K – kutya), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életkedve. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Pisti kedvének az egymás utáni napokon való alakulása: j – jó kedvű, a – átlagos, r – rosszkedvű. Feltehetjük, hogy a fájl formátuma helyes.

20. Kati **házi kedvencei** a tarantulák, az arany hörcsögök és a macskák. Az állatoknak az életerejük megőrzéséhez a táplálékon túl egyéb dolgokra is szükségük van: a tarantuláknak száraz és meleg terráriumra; az arany hörcsögöknek puha alomra, ahová befúrhatják magukat, a macskáknak rendszeres simogatásra. Kati állatainak van neve és ismerhető az életerejüket mutató 0 és 70 között szám (0 esetén az állat elpusztul). Katinak vannak jobb és rosszabb napjai. Mikor vidám, egyik állatáról sem feledkezik meg: ilyenkor a tarantulák életereje 1-gyel, a hörcsögöké 2-vel, a macskáké 3-mal nő. Átlagos napokon csak a macskáival foglalkozik (életerejük 3-mal nő), a többi állat életereje ilyenkor csökken: a tarantuláké 2-vel, a hörcsögöké 3-mal. Amikor szomorú, csak a legszükségesebb teendőket látja el és ezért minden állat egy kicsit gyengébb lesz: a tarantulák 3 egységgel, a hörcsögök 5-tel, a macskák 7-tel.

Kati kedve egy adott napon egy kategóriával jobb lesz attól, ha minden élő kedvencének az életereje legalább 5.

Naponként mutassa meg az állatok összes tulajdonságát, valamint nevezze meg az egyes napok végén a legéletrevalóbb állat(oka)t!

Az állatok adatait egy szöveges állományban találjuk. Az első sor tartalmazza az állatok számát, amelyet külön-külön sorban az állatok adatai követnek. Ebben egy karakter azonosítja az állat fajtáját (T – tarantula, H – hörcsög, M – macska), amit szóköz után az állat neve követ, majd újabb szóköz után a kezdeti életereje. Az állományban az állatok felsorolását követő utolsó sorban egy betű sorozat (sztring) írja le Kati kedvének az egymás utáni napokon való alakulása: v – vidám, a – átlagos, s – szomorú. Feltehetjük, hogy a fájl formátuma helyes.