

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 övezete (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 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 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 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 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 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 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. Egy **kamionos** 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, fülkés: 2). A kamionok vagy úton vannak, vagy az egyik telephelyen állnak.

- Melyik telephelyhez tartozik éppen most a legkevesebb kamion?
- 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.

- Volt-e olyan kamion, amely túl volt terhelve valamelyik fuvarban?
- 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 fuvar.

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 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 egy cég telephelyeit, vontatóit, sofőrjeit, fuvarjait. Készítsen teszteseteket, néhányat rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

5. Egy **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 elérhetőek, 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 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 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. Egy **teniszkлуб** 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 mondhaszon egy foglalást. Meg lehessen válaszolni az alábbi kérdéseket:

- Mennyi pályahasználati díjat kell fizetnie az adott napra egy adott tagnak?
- Mennyi a teniszkлуб aznapi összbevétele?
- Keressünk egy adott időpontra megadott borítású foglalható pályákat.
- 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 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 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!

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. Foglalás nélkül is lehet szabad helyet vásárolni. Egy jegy árára (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 állapotgépet! 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 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 filmszínház termeit, előadásait, foglalásait és jegyvásárlásait! Készítsen teszteseteket, néhánynak rajzolja fel a szekvencia diagramját, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!