

Szoftvertechnológia és – technikák

3. gyakorlat – osztálydiagram, kódgenerálás alapok

1. A gyakorlat menete

A gyakorlat folyamán komplexebb osztálydiagramos feladatok kerülnek megoldásra. A pluszpontos feladat keretein belül pedig betekintést nyerhetünk a kódgenerálás alapjaiba.

2. Vezetett feladatok

Építési vállalat

Készítsünk közösen osztálydiagramot Modelio-ban az alábbi szöveges leírás alapján!

Egy építési anyagokat forgalmazó vállalat telephelyét modellezük. A telephelyen raktárak vannak. Egy telephelyen maximum 10 raktár lehet, de legalább egy raktár mindig van a telephelyen. A raktárak lehetnek klimatizáltak. A raktárakban árukat tárolnak. Az árukról nyilvántartják azok egyedi leltári számát és tömegét, melyeket közvetlenül nem lehet módosítani. A raktárban a leltári szám alapján lehet lekérdezni az egyes áruk mennyiségét. Egyes áruk raklapokon tárolhatók. Minden raklapon tárolt árunak van egyedi módon kiszámítandó raklapterhelése (ami egy szám), viszont a nem raklapon tárolt áruknak nincs. Árukból jelenleg gerendát és zsalukövet tartanak nyilván. A gerendának három típusa van: vasbeton, kerámia, fa. Léteznek leárazott gerendák, amelyeket minden esetben raklapos áruként viselkednek. A zsalukövet mindig raklapokon tárolják. A zsalukövet raklapokon tárolják. A raktárakban raktárosok dolgoznak. Egy raktáros csak egy adott raktárban dolgozhat, ami az ő bázisraktára. A raktáros képes árukat leltárba venni és árut kiadni. A telephelyhez tartozik egy telepvezető, aki képes elvégezni egy raktáros minden feladatát. Ezen felül, a telepvezető le is tud selejtezni egy adott árut.

Webshop

Készítsünk közösen osztálydiagramot Modelio-ban az alábbi szöveges leírás alapján!

Egy webshop alkalmazáshoz készítünk osztálydiagramot. Egy felhasználó több webshophoz is kapcsolódhat, megadható, hogy ezek közül melyik a kedvence. A webshopban termékek vannak, amelyeknek van ára és azonosítója. A felhasználókról tároljuk a nevüket, születési dátumukat és irányítószámukat. A felhasználók kosárba gyűjthetik a vásárlásaik során a termékeket, egy-egy termékből többet is, de maximum 50db-ot. A felhasználók szabadon tudnak termékeket elhelyezni, ill. kivenni a saját kosarukból. A kosártól bármikor le lehet kérdezni a benne lévő termékek összesített árát. Ha a felhasználó végzett a vásárlással, a webshopnak elküldi a kosarát, amire az válaszként egy számlát küld. A felhasználó ki tudja fizetni a számlát, a fizetés eredményeképpen egy kódot kap, amit a webshopba visszaküldve befejezheti a vásárlást a szállítási mód megadásával (Postapont, Személyes átvétel, vagy Pick Pack pont). A felhasználó a számla fizetését, kétféleképpen rendezheti: kártyával, vagy ajándékkuponnal. Később más típusú fizetés bevezetése is lehetséges és biztonsági okokból fontos, hogy a fizetési módok megvalósítása egymástól teljesen független legyen. Kártyánál a fizetés folyamata során a kártyaszámot és CCV kódot, míg kuponnál csak a kuponkódot kell megadni. Az adatok megadása után a fizetés ellenőrizhető, majd kiegyenlíthető. A kiegyenlítés eredménye a kód, amit a webshopnak be kell küldeni.

3. Önálló feladatok

Szemétszállító cég – osztálydiagram készítése

Készítsen osztálydiagramot Modelio-ban az alábbi szöveges leírás alapján!

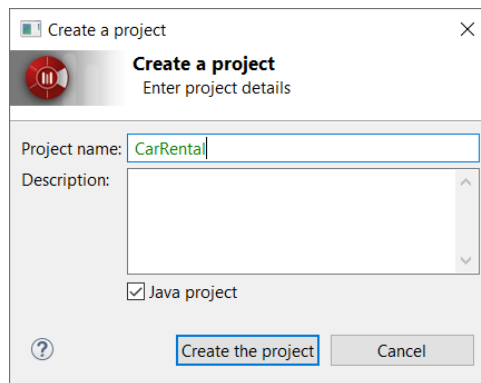
Egy személyszállító cég rendszerét modellezzük osztálydiagram segítségével. A cég alkalmazottairól tároljuk a nevüket és személyi számukat. Két féle alkalmazott dolgozik a cégben: szeméttűrítő és sofőr. A sofőrökről tároljuk a jogosítványuk nyilvántartási számát. A cégnek kukásautói vannak, amelyekhez egy sofőr és tetszőleges számú szeméttűrítő tartozhat. A sofőrökről is tároljuk, hogy melyik autót vezetik. A sofőrök az autókat a nap kezdetén felveszik a telephelyről, a nap végén pedig leadják. A telephelyen lehetőség van a kukásautók karbantartására. A karbantartás a következők egyike lehet: tisztítás, karosszéria javítás, ill. olajcsere. A rendszerben nyilvántartjuk a kiszolgált ingatlanokat. Az ingatlanról tároljuk a címét, illetve a hozzá tartozó kukákat, valamint azt, hogy melyik telephelyhez tartoznak. A szeméttűrítő képes a kukák kiürítésére. Az ingatlanok, telephelyek és a kukásautók GPS koordinátája (egyetlen string) a rendszerben egységesen kérdezhetőek le.

Autókölcshöz – osztálydiagram készítése és kódgenerálás (+ pontos feladat)

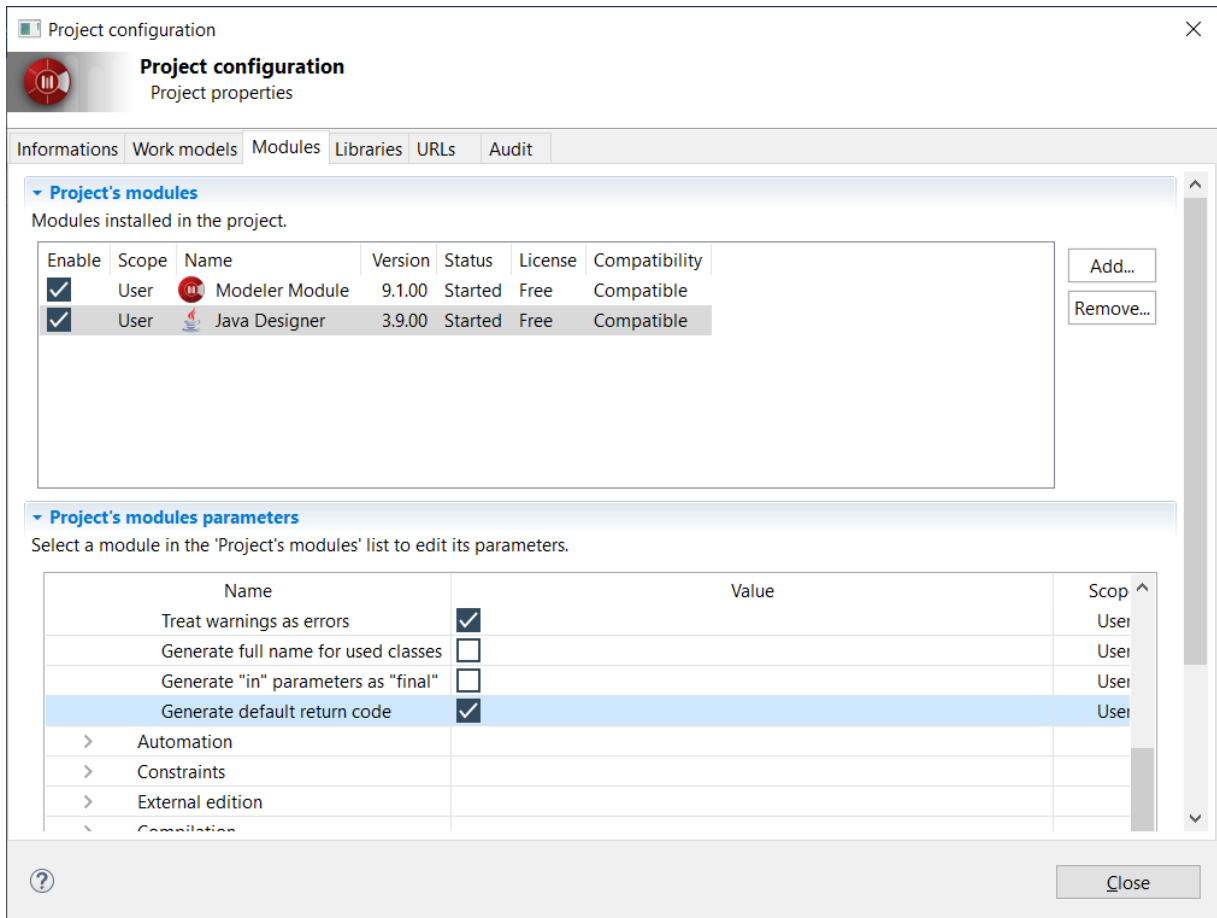
Készítsünk osztálydiagramot Modelio-ban az alábbi szöveges leírás alapján, majd generáljunk belőle Java kódot! **Beadandó:** Képernyőkép az osztálydiagramról és a generált Java források egy tömörített állományban.

Fontos megjegyzések a Modelio használatához ennél a feladatnál:

- File > Create Project ... > a projekt neve legyen *CarRental*
- **Pipáljuk be a Java project beállítást!**



- CarRental > CarRental > CarRental folderen jobb klikk, Create diagram... Class diagram
- Ha esetleg elfelejtettük bepipálni a Java project beállítást a Configuration menüpontból a projekt létrehozása után is hozzáadhatjuk a Java Designert: Configuration ->Modules->Add...->Válasszuk ki a Java Designer-t!
- Ellenőrizzük a Modules ablakban, hogy a Java Designer beállításainál be van-e pipálva a Generate default return code lehetőség.



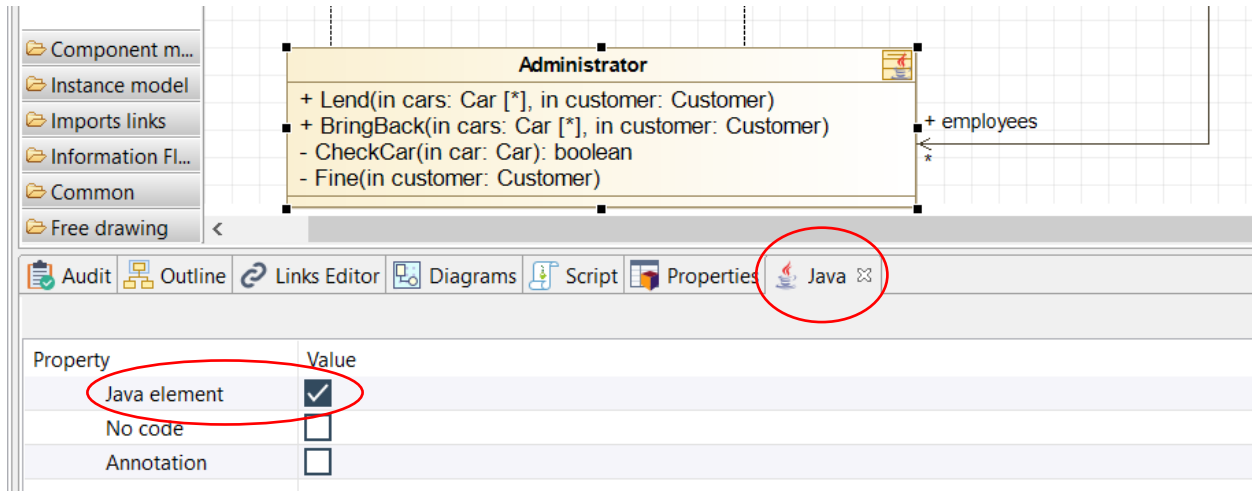
- Készítsük el az osztálydiagramot a szöveges specifikáció alapján!

FELADAT:

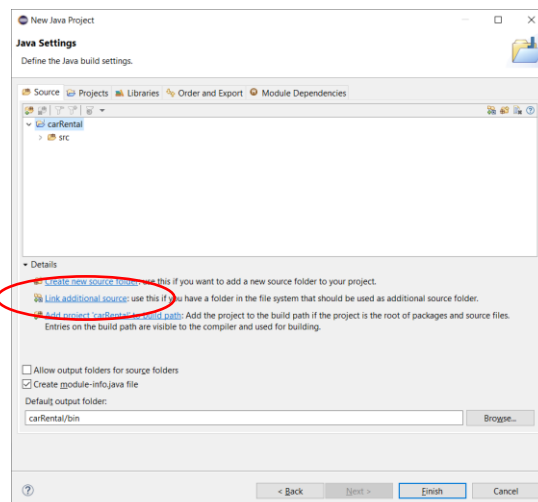
Egy autókölcsönző nyilvántartási rendszerét modellezük. Az autókölcsönzőben (**CarRental**) természetesen vannak autók (**Car**), amelyekről tároljuk a gyártóját, a rendszámát és az alvázszámát. Ezen kívül az autónak típusa (**CarType**) is van, ami lehet sedan, SUV, pickup vagy roadster. A gyártó és/vagy típus alapján képesek vagyunk az autókölcsönzőn keresztül autót keresni. Az autókölcsönzőnek vannak ügyfelei (**Customer**), akik autókat kölcsönözhetnek. Egy ügyfél egyszerre több autót is kölcsönözhet. Emellett az ügyfél elő is jegyezhet magának autót, de csak egyet. Az ügyfelekről tároljuk a személyi számát és a születési dátumát. Az ügyfélnek neve és címe (irányítószám, város, utca), ami lekérdezhető. Az autókölcsönzőnek van neve és vannak irodái, amelyeknek van címe. Az irodákban ügyintézők (**Administrator**) dolgoznak, akiken keresztül az ügyfelek képesek autókat kölcsönözni és visszahozni. E két művelethez kapcsolódóan az ügyintéző képes a kölcsönzés során ellenőrizni, hogy az autó kikölcsönzése lehetséges-e, illetve késedelmes visszahozatal esetén képes az ügyfelet megbüntetni.

- Generáljuk Java kódot az elkészült osztálydiagramból!
- Minden **osztályt, enumerációt, interfészt és csomagot** Java elemmé kell tennünk. Ezt az adott elem kiválasztásával, majd a Java fülre navigálásával tehetjük meg, ahol pipáljuk be a „Java element” beállítást. Az interfészeknél szükség lehet a modell frissítésére (Update model from the interface... felugró ablak). Ebben az esetben válasszuk a „Yes” lehetőséget. Így az interfész

metódusai is megjelennek az osztályokban, ami az osztálydiagramon redundanciát okoz, viszont a kódgenerálásnál így fognak helyesen bekerülni az egyes metódusok az osztályokba.



- A CarRental > CarRental> CarRental folderen jobb klikk >Java Designer >Generate. A forráskód ide generálódik: c:\Users\...\modelio\workspace\CarRental\src (**Fontos:** Ha mégsem találjuk a generált fájlokat ezen az elérési útvonalon, akkor használjuk a Help->Open Log File lehetőséget, ahol ki tudjuk keresni a valós elérési útvonalat!)
- Vizsgáljuk meg a generált kódot! Ne felejtsek el beadni a generált Java forrásfájlokat sem!
- Ezen a ponton válasszunk az alábbi két lehetőség közül:
 - Egyszerűen csak megnyitjuk Notepad++-ban a generált forrásokat és úgy vizsgáljuk meg őket.
 - Belinkeljük a forráskódot Eclipse-be. (Ennek a folyamatnak a megismerése hasznos lehet a jövőben, ha a generált kódot szeretnénk kiegészíteni. A generált kód kiegészítését értelemszerűen könnyebb elvégezni az Eclipse IDE-ben.)
- **Eclipse esetén:** File > New > Java Project (Ha nincs by default ott, akkor Project... és ott kikereshető)
- Projekt neve pl. carRental
- Next után a Source tabon lesz egy *Link additional source* link:



Szoftvertechnológia- és technikák – 3. gyakorlat

- Kattintás után ki kell browse-olni a foldert, ahova a Modelio generál: C:\Users\...\modelio\workspace\CarRental\src. Az eclipse-projekt beli nevet is adjuk meg, pl. modelio-src
- A generált forrásban lesznek modelio-s annotációk, amik a forrás és modell közti kapcsolatot definiálják. Hogy ezek is leforduljanak, a build path-ba be kell tenni egy jar fájlt, ennek módja:
- A kiinduló anyagok **Source** mappájában megtalálható egy javadesigner.jar fájl.
- Jobb klikk Eclipse-ben a projekten > Build Path > Configure build path > Libraries tab, Add External JARs, majd válasszuk ki a javadesigner.jar fájlt.