

Java asszociáció

Készítette: Goldschmidt Balázs, BME IIT, 2019.

1 Hallgatói példa letöltése

- Hozzunk létre egy új Java projektet *registry* néven!
- Hozzunk létre egy *registry* nevű csomagot a projektben!
- Töltsük le a hallgatói példa osztályainak forráskódját a tárgy honlapjáról, és a fájlokat másoljuk a projekt *src/registry* nevű mappájába!
- Fordítsuk le és futtassuk a programot!

2 Hallgatói példa kivételkezeléssel

Bővítsük a példát az előadáson tanult módon kivételkezeléssel!

- A *Course* osztályban adjunk kivételdobást az *addStudent* és a *removeByNeptun* metódusokhoz! Dobjanak “Kurzus megtelt” illetve “Nincs ilyen neptunkód” üzenetű kivételeket!
- A főprogramot bővítsük ki a kivételek elkapásával! Írassuk ki a kivétel üzenetét és a stacktrace-t!
- Módosítsuk úgy a főprogramot, hogy a kivételkezelés lefusson!
Adjunk túl sok hallgatót a tárgyhöz, illetve töröljünk olyan neptunkódút, amilyen nincs a tárgyhöz!
Futtassuk is a programot!
- Módosítsuk úgy a *Course* osztály *addStudent* metódusát, hogy dobjon kivételt, amikor olyan hallgatót adunk a tárgyhöz, aki már a tárgyat felvette!
Futtassuk is a programot!

3 Gépjárműnyilvántartás alaposztályai

Készítsünk gépjárműnyilvántartó programot!

Megvalósítandó osztályok:

- Vehicle (jármű)
 - attribútumok: gyártás éve (int), rendszám (String), típus (String)
 - metódusok: konstruktorok az attribútumokhoz, getter függvények, toString
- Person (személy)
 - attribútumok: születés éve (int), név (String), lakcím (String)
 - metódusok: konstruktorok az attribútumokhoz, getter függvények, toString
- Főprogram (Main osztályban)
 - hozzunk létre 2 autót és 2 személyt, írassuk ki őket a szabványos kimeneten!

4 Gépjárműnyilvántartás asszociációi

Bővítsük a gépjárműnyilvántartó programot az osztályok közötti kapcsolatokkal!

- Vehicle: lehessen tulajdonosa (*owner*). Ezt konstruktorban nem állítjuk, de van getter, setter metódusa.
- Person: lehessen járműve (*cars*), akár több is (ArrayList!), ehhez kapcsolódóan *addCar(Car c)* és *removeCar(String platenum)* metódusokkal, amelyek hozzáadnak egy meglevő autót, illetve törölnek a listából a rendszám alapján egyet. Mindkettő állítsa az autó tulajdonosát is! (Törléskor a tulajdonos legyen null.)
- Person: lehessen a személyhez rendelt autókat listázni (*listCars()*).
- Próbáljuk ki a programot! Adjuk hozzá a létrehozott járműveket az egyik személyhez, majd listáztassuk ki a személyek autóit!

5 Gépjárműnyilvántartás kivételkezeléssel

Bővítsük a gépjárműnyilvántartó programot kivételkezeléssel!

- Person: ha olyan rendszámmal szeretnénk törölni, amilyen nincs az illetőnek, dobjon kivételt („hibás rendszám”).
- A főprogramban próbáljunk hibás rendszámot törölni, és kapjuk el a keletkező kivételt!

6 Gépjárműnyilvántartás adatfeldolgozással

- Az autókat és a személyeket tároljuk egy-egy ArrayListben! Figyeljünk az ArrayList helyes paraméterezésére (ArrayList<Vehicle> és ArrayList<Person>).
- Írjunk (statikus) metódust (*oldCars*), amely egy Vehicle-öket tartalmazó ArrayListet kap paraméterül, és visszaadja a szintén paraméterként megkapottnál idősebb autók számát!
- Írjunk (statikus) metódust (*oldOwners*), amely egy Person-öket tároló ArrayListet kap paraméterül, és visszaadja a 10 évnél régebbi autóval rendelkező tulajdonosok számát! Használjuk fel az *oldCars* metódust!
- Írassuk ki a szabványos kimenetre a 10 évnél idősebb autóval rendelkezők számát!

7 Gépjárműnyilvántartás listakezeléssel

- Módosítsuk az *oldOwners* metódust: ne a számosságot adja vissza, hanem egy ArrayListben a tulajdonosokat, akiknek 10 évnél régebbi az autójuk van.
- Írassuk ki a 10 évnél régebbi autóval rendelkezők nevét!