

# Java öröklés

---

*Készítette: Goldschmidt Balázs, BME IIT, 2019.*

## 1 Hallgatói példa (örökléses változat) letöltése

- Hozunk létre egy új Java projektet *registry2* néven!
- Hozunk létre egy *registry2* nevű csomagot a projektben!
- Töltsük le a hallgatói példa osztályainak forráskódját a tárgy honlapjáról, és a fájlokat másoljuk a projekt *src/registry2* nevű mappájába!
- Fordítsuk le és futtassuk a programot!

## 2 Láthatóságok

Próbáljuk ki a különböző, láthatóságot állító módosítókat.

- Hozunk létre egy *whatever* nevű, paraméter és visszatérési érték nélküli publikus metódust a *Student* osztályban!
- A *whatever* metódus módosítsa a *Student* *job* attribútumát 2000-re, a *credits*-et 30-ra. Mit tapasztalunk?
- A *Person*-ben módosítsuk a *job* láthatóságát *protected*-re, Most mit tapasztalunk a *whatever*-ben?
- Mikor tesszük *public*, *protected* és *private* elérésűvé az attribútumokat?

## 3 Újabb *Person* leszármazottak

Hozunk létre újabb leszármazottat a *Person*-ből!

- Hozunk létre egy *Artist* nevű osztályt, ami művészeket fog reprezentálni.
- Az osztály származzon a *Person*-ből, de legyen egy *String* típusú, *art* (művészeti ág) nevű attribútuma is, amit kívülről nem lehet elérni, de egy metódussal le lehet kérdezni. Csak konstruktorból legyen állítható.
- Az osztály *toString* metódusa is jelenítse meg a művészeti ágat!
- A főprogramban vegyünk fel néhány művészt egy *ArrayList*-be és írassuk is ki őket!
- Extra feladat:* Írjunk metódust a *Person* osztályba (*meet*), ami paraméterként egy másik *Person*-t kap, és a saját *greetings()* metódusából és a paraméterként kapott *Person* nevéből képezve kiír egy üdvözetet, pl: "Üdvözetem! Gipsz Jakab!". Próbáljuk is ki két különböző objektummal!

## 4 Hallgatói példa (örökléses változat) kivételkezeléssel

Módosítsuk a példát az előadáson tanult módon specializált kivételkezeléssel!

- a) Hozunk létre három új kivételosztályt (*CourseException*, *CourseFullException* és *NeptunNotFoundException*)! Az első legyen az *Exception* leszármazottja, a második kettő a *CourseException* leszármazottja!
- b) A *CourseException* osztály a konstruktorában vegye át a kurzust, amihez kapcsolódva dobódhat! A *getMessage()* metódus legyen felüldefiniálva, és adja vissza a kurzus nevét!
- c) Módosítsuk az előadáson látott módon a *Course* osztály *addStudent* és *removeByNeptun* metódusait, de a kivételek a konstruktorban kapják meg az érintett kurzus referenciáját is!
- d) Próbáljuk ki, hogy csak a *CourseException*-t kapjuk el!
- e) Próbáljuk ki, hogy a *CourseException* helyett a specializált kivételeket kapjuk el! Ehhez a *try* blokk után legyen több *catch* ág is, és egyik esetben "Tele van", másik esetben "Nincs ilyen" szöveget is írassunk ki!

## 5 Leszármazottak leszármazottai

Hozunk létre leszármazottat az *Artist*-ből!

- a) Módosítsuk úgy az *Artist* osztályt, hogy ne lehessen belőle példányt létrehozni! Ezt próbáljuk egyetlen kulcsszó hozzáadásával elérni!
- b) Származtassunk le az *Artist*-ből néhány művész-típust: *Painter*, *Singer*, *Writer*, stb.
- c) A leszármazottak a konstruktorukban az *art*-ot állítsák be a típusuknak megfelelően: *Painter* - „festészet”, *Singer* - „zene”, *Writer* - „irodalom”, stb.
- d) A főprogramban vegyünk fel különböző művészeket, és írassuk is ki őket a szabványos kimenetre!
- e) Készítsünk egy önálló, nem a *Person*-ből, nem az *Artist*-ből, stb származó művészt (*Drummer*), aki minden olyan tulajdonsággal (publikus metódusok, konstruktorok, stb.) rendelkezik, amivel a *Singer* is!
- f) Hány érdemi (nem üres) kódsorral hosszabb a *Drummer* kódja, mint a *Singer*-é?