

1. feladat: Rövid kérdések**15 pont**

1.1 Egy újonnan fejlesztett adatstruktúra memóriaigényét mérjük. Az alkalmazást Java nyelven fejlesztettük. A mérés során a következő megfigyeléseket tettük:

- 1 GB memóriakorláttal minden mérést el tudunk végezni, de magas futásidőket mérünk a Java szemétgyűjtő (garbage collection) mechanizmusa miatt.
- Másrésről, ha 32 GB memóriával futtatjuk a méréseket, akkor alacsony futási időket mérünk, de magas memóriahasználatot (a használt profilozó szerint ez 22 GB memóriahasználatot jelent).

Magyarázza el a mérésekben tapasztalt ellentmondást, és javasoljon egy módszert a memóriaigény mérésére!

1.2 Új adatstruktúránk memóriaigényét öt egyre nehezebb probléma méretén értékeltük ki 10-szer, és az eredményeket **pont-pont diagramon (scatterplot)** szeretnénk megjeleníteni. Rajzoljon egy példát egy **pont-pont diagramra** öt különböző probléma méretre! Mi történik a pont-pont diagramokkal, ha több mérésnek ugyanazok az értékei?

1.3 Mutasson egy példát egy **vezérlési folyam diagramra** és néhány **teszt bemenetre**, ahol az utasítás fedettség 100% (statement coverage), de az MC/DC lefedettség nem!

2. feladat: Gráf alapú modellezés**25 pont**

a) Készítsen egy **Refinery metamodellt** az alábbi specifikáció alapján:

Egy ókori szövegek rekonstrukciós és fordítási munkafolyamatát kezelő rendszert szeretnénk modellezni. Minden ókori szövegtöredék legfeljebb 1 múzeumi gyűjteményben lehet tárolva. Egy rekonstrukció 1-100 töredéket egyesít, és három állapot egyikében lehet: javasolt, jóváhagyott vagy vitatott. Egy fordítási projekt pontosan egy elfogadott rekonstrukción dolgozik és legalább 1 fordítási segédeszközt használ. A szótárak olyan fordítási segédeszközök, amelyek pontosan 2 nyelvjáráshoz tartoznak. A MI-modellek olyan fordítási segédeszközök, amelyeket legalább 1 nyelvjáráson tanítottak be. Ezek lehetnek rekurrens neurális hálózatok vagy transzformerek. Minden nyelvjárást legalább 1 kutató tanulmányoz, és minden fordítási projektet legalább 2 kutatónak kell szakmailag lektorálnia. Tudjuk még továbbá, hogy minden múzeumi gyűjteményt pontosan 1 kutató gondoz.

Kizárólag Refinery kódot írjon! NE írjon Java kódot vagy rajzoljon UML osztálydiagrammot.

Használja az alábbi fogalmakat: AcceptedReconstruction (elfogadott rekonstrukció), AIModel (MI-modell), curatedBy (gondozója), Dialect (nyelvjárás), dialects (nyelvjárásai), Dictionary (szótár), DisputedReconstruction (vitatott rekonstrukció), fragments (töredékei), MuseumCollection (múzeumi gyűjtemény), ProposedReconstruction (javasolt rekonstrukció), Reconstruction (rekonstrukció), RecurrentNeuralNetwork (rekurrens neurális hálózat), reviewedBy (lektorai), Scholar (kutató), storedIn (tárolója), studiedBy (tanulmányozói), TextFragment (szövegtöredék), trainedOn (tanult rajta), Transformer (transzformer), TranslationAid (fordítási segédeszköz), TranslationProject (fordítási project), uses (használja), worksOn (dolgozik rajta)

b) Rajzoljon **gráfmodellt** az alábbi adatok alapján:

A Drezdai Kódex B3, B4 és B5 töredékeit a Drezdai Állami Művészeti Gyűjteményben őrzik. Ezek részei egy elfogadott rekonstrukciónak, amelyet „Drezdai Almanach Szekció”-nak neveznek. A Drezdai Időjárési Szakaszok fordítási projekt ezen a rekonstrukción dolgozik egy MayaGPT nevű transzformer MI-modell segítségével, amelyet klasszikus maja és ch'olti' nyelvjárásokon tanítottak be. A projekt emellett egy klasszikus maja – gyarmati spanyol szótárt is használ. A projektet két kutató lektorálja: Prof. Lisa Chen (a drezdai gyűjtemény gondozója) és Dr. Maria Rodriguez. Mindkét lektor tanulmányozza a klasszikus maját és a ch'olti' nyelvjárást. Dr. Rodriguez a gyarmati spanyolt is tanulmányozza.

Kizárólag gráfmodellt rajzoljon! NE írjon Java vagy Refinery kódot.

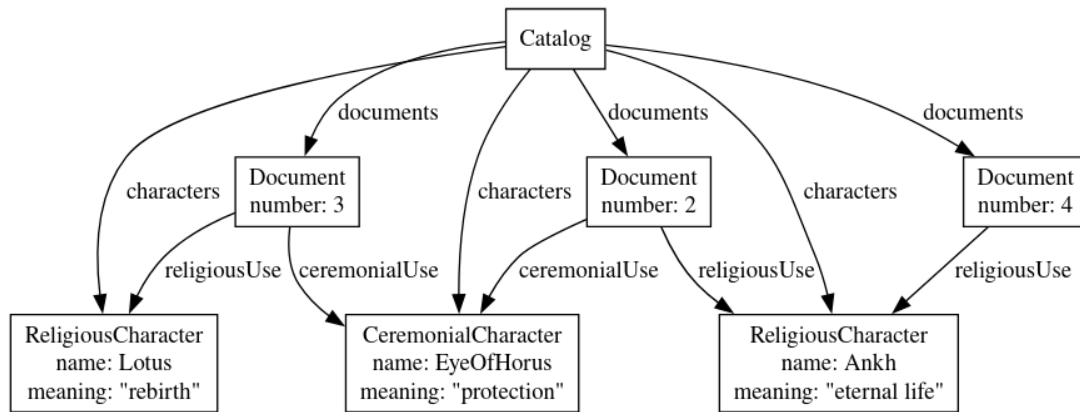
c) Egy szakértő szerint a következő **jólformáltsági kényszer** érvényes: *egy lektornak legalább egy, a fordítási projektben szereplő nyelvjárást tanulmányoznia kell.* A szakértő azonban gyanítja, hogy ezt a metamodell jelenlegi verziójával nem lehet kikényszeríteni. Ennek megerősítéshez rajzoljon egy olyan **gráfmodellt**, amely megfelel a metamodellnek, de megsérti a kényszert!

d) A szakértő szerint egyes fordítási projektek akár javasolt vagy vitatott rekonstrukciókat is használhatnak. **Javasoljon** egy módszert a metamodel módosítására, hogy az ilyen kéziratokat is kezelhessük.

3. feladat: Szöveges modellezés

30 pont

Egy **szöveges szakterületspecifikus nyelvet** szeretnénk készíteni, hogy megadhassuk a hieroglif írás karaktereinek jelentéseit és előfordulásait. Az alábbi példa a kívánt **konkrét szintaxist** (szöveges leírást) és **absztrakt szintaxist** (példánygráfot) mutatja be a nyelvhez:



characters:

- Lotus means "rebirth" and is a religious character
- Ankh means "eternal life" and is a religious character
- EyeOfHorus means "protection" and is a ceremonial character

documents:

- Tablet 2 uses Ankh and EyeOfHorus
- Tablet 3 uses Lotus and EyeOfHorus
- Tablet 4 uses Ankh

a) Készítsen egy **Langium nyelvtant** a nyelv elemzésére! Az alábbi deklarációk már rendelkezésére állnak:

grammar Hieroglyphic

```
hidden terminal WS: /\s+;/
terminal ID: /[_a-zA-Z][_w_]*;/
terminal INT: /\d+;/
terminal STRING: /"[^"]*"//;
```

Adja meg a nyelvtan hiányzó részét!

b) Készítsen **Jinja2 sablont** a karakterek előfordulásai relációs adatbázisba beszűrő SQL szkript generálásához! A sablon bemenete az a **Catalog** objektum, amelyet az a) részben létrehozott nyelvtani elemzővel olvastunk be. Egy példa SQL szkript az alábbiakban látható:

```
INSERT INTO characterUses (documentId, character, meaning)
VALUES
(2, 'Ankh', 'eternal life')
(2, 'EyeOfHorus', 'ceremonial')
(3, 'Lotus', 'rebirth')
(3, 'EyeOfHorus', 'ceremonial')
(4, 'Ankh', 'eternal life')
```

A könnyebb olvashatóság kedvéért a közvetlenül a példánymodellből származó szövegeket **félkövérrel** emeltük ki. A szertartási karaktereknél a példánymodellben szereplő érték helyett a 'ceremonial' jelentést adja meg a szkriptben. Feltételezhetjük, hogy minden karakter neve és jelentése egy érvényes SQL sztring, amely nem tartalmazza a ' (aposztróf) speciális karaktert.

A **type** attribútum tartalmazza az objektumok típusát (pl. a `x.type == "ReligiousCharacter"` használatával ellenőrizheti, hogy az `x` **ReligiousCharacter** típusú-e) a példánymodellben. A kereszthivatkozások sztringként vannak tárolva, amelyek értékei megegyeznek a hivatkozott objektum nevével. A nem beállított kereszthivatkozásnál a megfelelő sztring nincs benne a példánymodellben.