

Kedves Dávid!

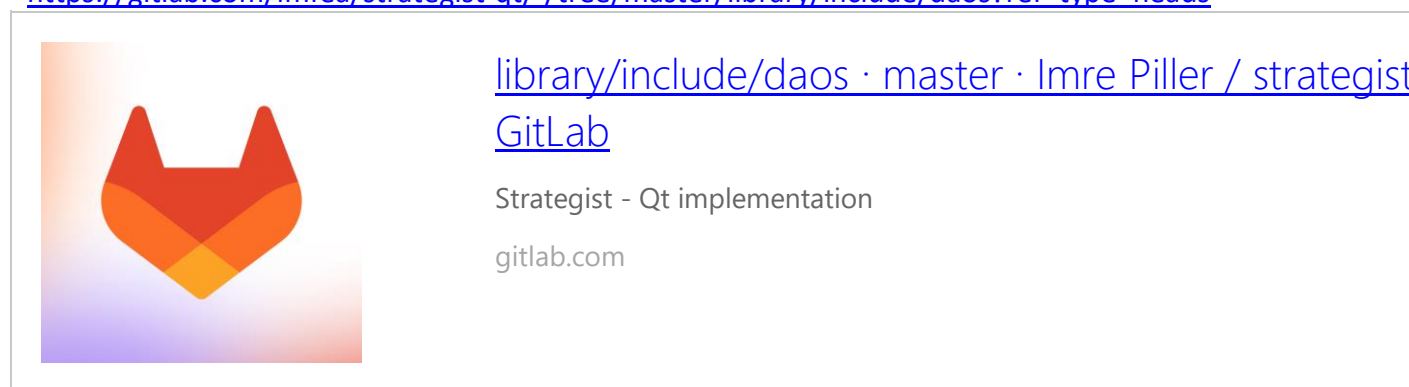
Nem írt még Fegyverneki Sándor (csak valami olyasmit, hogy majd írásban, de az még kiderül, hogy mire is gondolt pontosan).

Az osztálydiagram valamilyen hasonló kell, hogy legyen.

- OOP esetében érdemes meggondolni, hogy milyen származtatási lehetőségek vannak (illetve, hogy majd ez hogyan illeszthető JPA-hoz).
- Az adatok tárolásán kívül meg kellene gondolni, hogy milyen osztályok kellhetnek még az alkalmazáslogikához. Csak funkcionalitás szerint például valahogy meg kell majd oldani a PDF-ek létrehozását, e-mail-ek küldését.
- Java esetében konvenció szerint a metódusok neve kis betűvel kezdődik. Az aláhúzásjelek kerülendők. (Itt nem muszáj még konkrét programozási nyelvhez kötődni, de segíti a későbbi munkákat, hogy ha már most úgy van.)
- A típusoknál szintén szerepelhetnek már a Java-s típusok (például text helyett String).
- A metódusok neve jó, hogy ha mindig igével kezdődik (például Registration -> register, ZV_Create -> createExam, ThesisUpload -> uploadThesis).
- A paramétereket is meg kellene adni.
- Magának a ZV-nek is érdemes lehet külön osztályt definiálni (például Exam, Examination vagy Defense néven).

Egy DAO-s példa itt található:

- https://gitlab.com/imfea/strategist-qt/-/tree/master/library/include/daos?ref_type=heads



(Osztálydiagramot nem rajzoltam hozzá, de remélem, hogy a szerkezet alapján látszik, hogy miről lenne szó.)

> A CRUD-test-el kapcsolatban az lenne a kérdésem, hogy Tanár úr pontosan mire gondolt ezalatt?

A CRUD tesztek az osztálydiagram alapján lehet összegyűjteni. A Reviewer esetében például

- testRegister_Successful
- testRegister_MissingData
- testRegister_ExistingEmail
- testRegister_WeakPassword

Elnevezési konvenciókra csak ajánlásokat találtam, például:

<https://medium.com/@stefanovskyi/unit-test-naming-conventions-dd9208eadbea>

> Például amikor felviszek egy rekordot a webes felületen lévő formból az adatbázisba azt egyszerre 3 függvény valósítja meg java-ban, amelyből az érdemi void.

A tesztel nem feltétlenül a visszatérési értéket kell vizsgálni.

- Hogy ha egy létrehozás műveletről van szó, akkor meg lehet nézni, hogy az elemek száma nőtt-e, illetve hogy ha a visszatérési érték a létrehozással kapott elem azonosítója (amit célszerű úgy megoldani), akkor le lehet kérdezteni az elemet, és meg lehet vizsgálni, hogy a felvitelnél szereplő, és a lekérdezéssel kapott adatok megegyeznek-e.
- Hibás adatfelvitel esetén a tesztben az is szerepelhet, hogy kivételt dob a hívás.

> Tanár úr tudna egy példát küldeni esetleg, hogy mire gondolt?

Egység tesztekre itt van néhány példa:

- https://gitlab.com/imfea/exprail-python/-/tree/master/tests?ref_type=heads
- https://gitlab.com/imfea/strategist-qt/-/tree/master/tests?ref_type=heads
- https://gitlab.com/imfea/strategist-web/-/tree/master/server/tests?ref_type=heads

Üdvözlettel,
Piller Imre