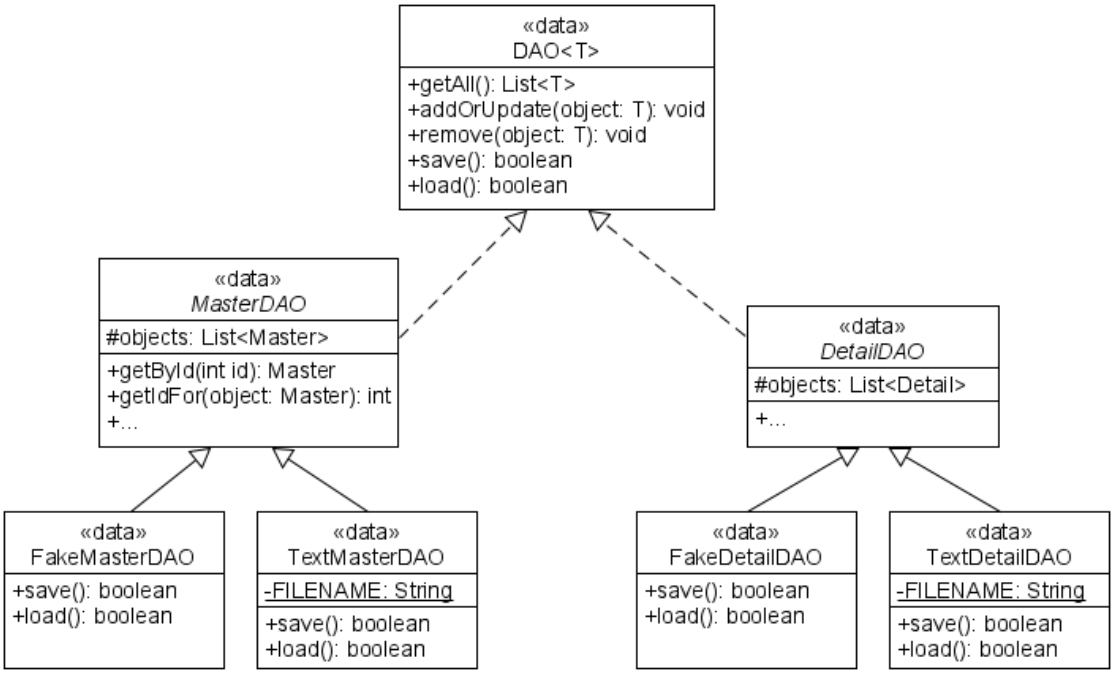
OOP2 – Practicumopdracht

Week 5

Vorige week heb je met gebruik van het *Data Access Object Design Pattern* (DAO) een laag aan je applicatie toegevoegd die de dataverwerking gaat verzorgen.

Dit was best een pittige klus om neer te zetten, met als resultaat dat het eigenlijk nog niets daadwerkelijk kon opslaan of inladen. Gelukkig gaan we daar deze week verandering in brengen.



Afbeelding: UML met twee nieuwe DAO-classes, er zijn geen veranderingen in de structuur nodig.

Werk nu het volgende stappenplan af met gebruik van de bovenstaande UML:

1. In tegenstelling tot je FakeDAO's van vorige week, kun je in de TextDAO's geen *hoortBij*-referentie meer hardcoden. Implementeer daarom de *getIdFor*-methode in de MasterDAO zodat deze de index van het object in de *objects*-List returned. Indien het gevraagde object niet bestaat, return dan -1.

2. Maak voor beide models twee nieuwe DAO-classes in de *data*-package en geef deze een naam in het volgende formaat:

   Text<NaamVanModel>DAO (bijv. *TextTodoLijstDAO* en *TextTodoRegelDAO*)

3. Implementeer voor beide DAO's de *load*- en *save*-methodes met gebruik van Text IO, je mag zelf bepalen op wat voor manier je dit doet en hoe de tekstbestanden eruit zien als je deze bijvoorbeeld bekijkt in een tekstbewerker. Uiteraard moet alle data daadwerkelijk opgeslagen en weer ingeladen kunnen worden.

    - **Let op!** De TextMasterDAO mag enkel Master-objecten opslaan, de TextDetailDAO enkel Detail-objecten. Maak daarom gebruik van de *getIdFor* in de MasterDAO om de *hoortBij*-referentie op te kunnen opslaan.

    - Zorg ervoor dat gebruikte bestanden worden vrijgegeven als je klaar bent.

    - Handel foutmeldingen die optreden af en print deze met *System.err* naar de console, zorg ervoor dat foutmeldingen specifiek aangeven wat er aan de hand is (bijv. "Bestand niet gevonden", "Bestand al in gebruik", enzovoorts).

4. Pas de *MainApplication* zodanig aan dat de rest van de applicatie gebruik maakt van de nieuwe DAO's.

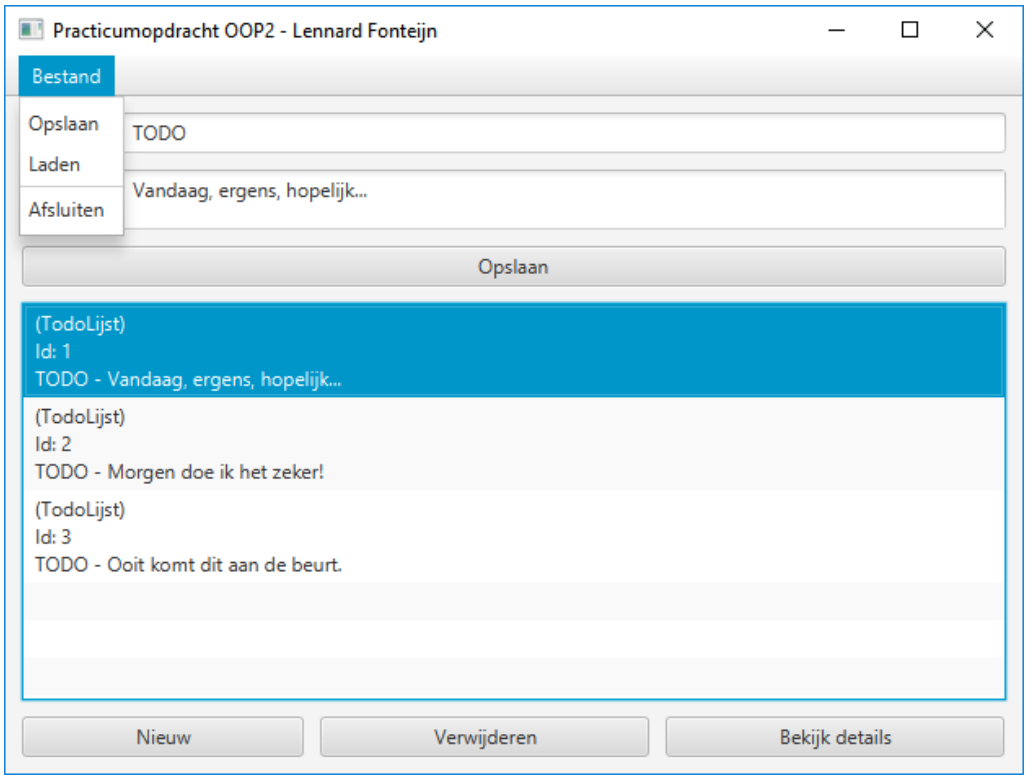
5. Breid het scherm behorende bij je Master-model uit met een menubalk, gebruik hierbij onder andere een *BorderPane*. Zorg ervoor dat je menubalk een optie bevat voor "Bestand", met daarin drie andere opties: "Laden", "Opslaan" en "Afsluiten".

6. Zorg ervoor dat de menu-opties als volgt gaan werken:

    - Wanneer de gebruiker op "Laden" klikt, moet op beide DAO's de *load*-methode worden aangeroepen. Vraag de gebruiker om toestemming met een Ja/Nee-*Alert* voordat je dit daadwerkelijk doet. Laat de gebruiker vervolgens met een *Alert* weten of het laden is gelukt of niet. Zorg er tevens voor dat de ListView direct de ingeladen data toont.

    - Wanneer de gebruiker op "Opslaan" klikt, moet op beide DAO's de *save*-methode worden aangeroepen. Vraag de gebruiker om toestemming met een Ja/Nee-*Alert* voordat je dit daadwerkelijk doet. Laat de gebruiker vervolgens met een *Alert* weten of het opslaan is gelukt of niet.

    - Wanneer de gebruiker op "Afsluiten" klikt, moet de applicatie direct afsluiten. Vraag de gebruiker wel of de data nog eenmalig opgeslagen moet worden met een Ja/Nee-*Alert*, zodat er niet per ongeluk data verloren kan gaan. Voorkom hierbij tevens dubbele code.



Afbeelding: Voorbeeld van een TodoLijstView na het afronden van alle stappen.

Checklist

- ❑ De MasterDAO heeft een *getIdFor*-methode.
- ❑ De DAO's zijn geïmplementeerd zoals aangegeven in de UML.
  - ❑ Er is een TextMasterDAO- en TextDetailDAO-class aanwezig.
  - ❑ Beide classes zitten in een *data*-package.
- ❑ De DAO's vertonen het gedrag zoals beschreven in het stappenplan:
  - ❑ De *save*-methode slaat een tekstbestand op, geeft deze na gebruik vrij en handelt foutmeldingen af met een specifieke rede zichtbaar in de console.
  - ❑ De *load*-methode laad een eerder opgeslagen tekstbestand in, geeft deze na gebruik vrij en handelt foutmeldingen af met een specifieke rede zichtbaar in de console.
  - ❑ De TextMasterDAO slaat enkel Master-ojecten op, de TextDetailDAO slaat enkel Detail-objecten op.
  - ❑ Er is gebruik gemaakt van de *getById*- en *getIdFor*-methodes in de TextDetailDAO.
  - ❑ Alle data wordt daadwerkelijk opgeslagen en weer ingeladen.
- ❑ De applicatie maakt gebruik van de TextMasterDAO en TextDetailDAO.
- ❑ De Master-view is uitgebreid met een *BorderPane* met daarin een menubalk en het scherm zelf.
  - ❑ De menubalk bevat een optie "Bestand".
  - ❑ De optie "Bestand" bevat drie andere opties: "Laden", "Opslaan" en "Afsluiten".
- ❑ De menu-optie "Laden" werkt als volgt:
  - ❑ Er wordt een Ja/Nee-*Alert* getoond, bij de keuze "Nee" gebeurd er niks.
  - ❑ Data wordt ingeladen met gebruik van de *load*-methodes.
  - ❑ De status van het inladen wordt getoond aan de gebruiker met een *Alert*.
  - ❑ De *ListView* wordt bijgewerkt.
- ❑ De menu-optie "Opslaan" werkt als volgt:
  - ❑ Er wordt een Ja/Nee-*Alert* getoond, bij de keuze "Nee" gebeurd er niks.
  - ❑ Data wordt opgeslagen met gebruik van de *save*-methodes.
  - ❑ De status van het opslaan wordt getoond aan de gebruiker met een *Alert*.
- ❑ De menu-optie "Afsluiten" werkt als volgt:
  - ❑ Er wordt een Ja/Nee-*Alert* getoond, bij "Nee" wordt er niks opgeslagen.
  - ❑ Code voor opslaan wordt hergebruikt.
  - ❑ De applicatie sluit af.
- ❑ Het project bevat geen compile-errors.
- ❑ Het project is gecommit en gepushed naar GitLab.