

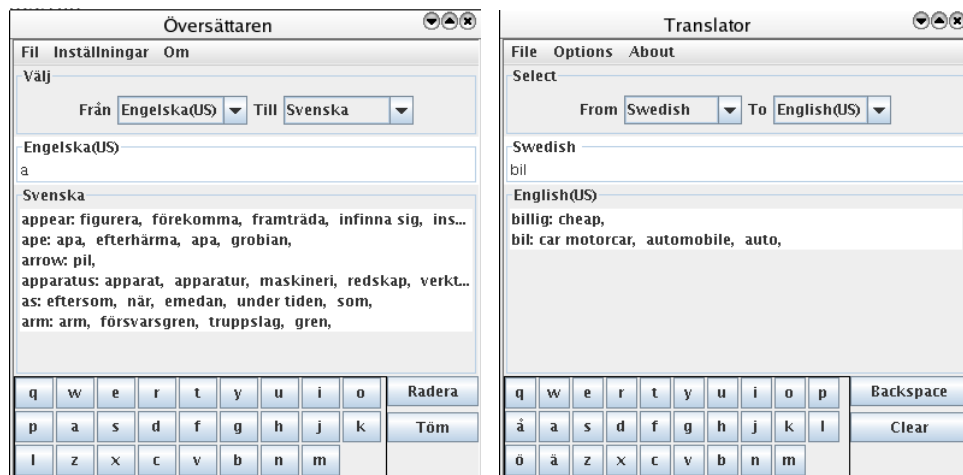
Laboration 2: En översättare

1 Syfte

- Fortsättning på testdriven utveckling, programmeringsprinciper och designmönster.
- Konstruktion av applikationer med grafiska användargränssnitt (MVC-modellen)

2 Uppgift

Vi skall nu använda det vi utvecklade i laboration 1 för att på så sätt skapa ett helt fungerande program. Vi förutsätter pekskärm (som simuleras m.h.a. musen), d.v.s. tangentbord behöver inte fungera. Nedan visas engelska till svenska med svenskt GUI (engelskt tangentbord) samt svenska till engelska med engelskt GUI (svenskt tangentbord).



3 Funktionalitet

Grundfunktionaliteten är oförändrad, finns angiven i del 1. Vissa funktionalitet har med I18N att göra, se bild ovan och vidare nedan.

- view innehåller alla GUI-klasser och någon hjälpklass, se kommentarer i dessa samt vidare nedan. GUI-klasserna använder IObservableTranslator för att registrera sig som observatörer och göra anrop på modellen.
- il8n- och exception-paketen ansvarar för respektive, internationalisering och felhantering.
- util innehåller en generell hjälpklass.

5.1 MVC

Applikationen skall byggas enligt MVC-modellen men utan separat C-del eftersom applikationen är så pass enkel (vår C-del består bara av lyssnare). V-delens anrop kommer att hamna i C-delen som i sin tur gör anrop på metoderna i M-delens gränssnitt (vilka eventuella levererar returvärden). Vid tillståndsförändringar i modellen används Observer-mönstret för att uppdatera V-delen. Alla V-delar som skall uppdateras måste alltså registrera sig som observatörer.

Implementering av Observer mönstret sker m.h.a. färdiga Java klasser och gränssnitt (PropertyChangeSupport, PropertyChangeListener, m.fl.).

Sammanfattning Förändringar av GUI:et kan ske på tre sätt;

- I lyssnaren (controllern) före anropet på modellen (lyssnaren ligger i GUI-klassen så alla komponenter är åtkomliga).
- I lyssnaren efter anropet på modellen.
- Modellen signalerar m.h.a. Observer-mönstret att tillståndet har förändrats. Förändringarna görs då i vyklassens callback-metod (metoden `propertyChange()`).

5.2 Människa datorinteraktion

Applikationen skall bete sig på ett människovänligt sätt, d.v.s. orimliga val skall justeras till rimliga. Text skall programmet smidigt hantera om man väljer samma från- och tillspråk m.m.

Tangentbordet anpassas till utgångsspråket, byter man till svenska skall tangenterna för å, ä och ö finnas. Menyn File har bara Exit som val och Menyn About visar upp en dialogruta.

Under menyn Inställningar (Options) kan man byta språk för GUI:et.

6 Metod

Övergripande kommer vi att göra följande;

1. Ta fram gränssnittet ITranslator och därefter implementera Translator. Testa.
2. Bygg på med observermönstret. Testa.

3. Koppla på GUI.
4. Utöka så att man kan byta språk i GUI:et (I18N).

7 Utvecklingsmiljö

Uppgift 1 Ett projektskelett finns färdig på kursidan. Hämta hem. Skall gå att köra.

För att kunna använda DictionaryFactory och övriga klasser från lab 1 måste man lägga till dictionary-projektet under Build Path (Build Path > Configure Build Path > Projects > Add).

8 Translator

Uppgift 1 Vilka metoder vill klasserna i view-paketet kunna anropa? In-utdata? Lägg till metoder i ITranslator. Här finns flera olika alternativ, tänk på att data även kan nå GUI:et m.h.a. observer, se vidare nedan.

Uppgift 2 Implementera Translator. Vilken data ansvarar klassen för? Klassen använder DictionaryFactory för att få tag i ett lexikon och ExceptionHandler för fel (finns färdig). Skriv en metod i taget och testa.

Eventuellt kan man göra en s.k. kommandoradsversion, d.v.s. hela applikationen utan GUI. Finns färdig klass CommandLineTranslator.

9 Observer

Vi skall nu bygga på Translator med observer-funktionalitet. Vi skall undvika att ändra i tidigare skriven kod (Open-Closed principle).

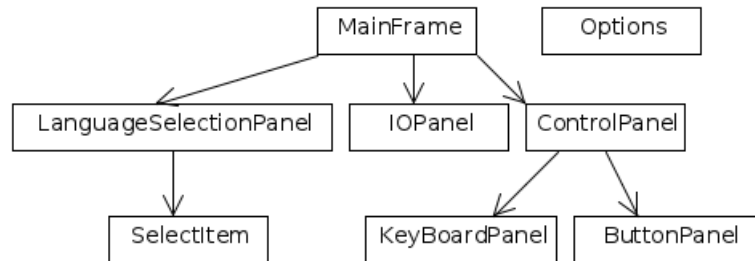
Uppgift 1 Inspektera den färdiga enum:en StateChanges och IObservableTranslator. Skapa en klass ObservableTranslator som implementerar IObservableTranslator. Låt klassen vara en Singleton. Klassen skall återanvända och utöka Translator så att observatörer blir informerade om tillståndsförändringar (i Translator). Klassens enda ansvar är alltså att hantera observer-delarna. Finns flera sätt att göra detta på. Obs! Att en metod kan behöva signalera flera olika StateChanges. Gör klart metoderna från gränssnittet och testa.

Testmetodik Det finns en påbörjad testklass TestObservableTranslator använd denna.

10 MVC

Nu skall vi koppla ihop GUI:et (som innehåller C-delarna) med modellen. GUI:et är uppbyggt av en JFrame (MainFrame) och ett antal paneler. SelectItem är en hjälpklass, används av ComboBoxar (i GUI:et visas en sträng men när vi väljer

skicka en annan sträng). Options håller bl.a. defaultvärden, se koden.



Uppgift 1 Inspektera GUI-klasserna. Alla observatörer måste registrera sig. Låt detta ske i konstruktorn. Vissa GUI-klasser fungerar som lyssnare, i dessa skall vi anropa modellen.

TIP: All intressant kod finns i metoderna `propertyChange` (data från modell till GUI, del av `Observer`) och `actionPerformed` (anrop från GUI till modell (eventuell returdata)).

Uppgift 2 Implementera den slutgiltiga funktionaliteten. F.n. kan vi inte automatisera detta utan vi får testa genom att köra.

11 I18N

I i18n-paketet finns en del kod för att göra GUI:et lokaliserbart (byta mellan engelska och svenska). GUI texterna ligger i properties-filer. Se klassen `KeyboardPanel` för användning av properties-filer tillsammans med `ResourceBundle`. Språkbyte för GUI görs som sagt under menyn `Options`. Se också klassen `Options`.

Uppgift 1 Skapa ett interface, `ILocalizable`, som alla lokaliserbara klasser måste implementera.

Uppgift 2 Skapa ett klass `Localizator`, som håller reda på alla lokaliserbara klasser. Låt klassen enbart ha statiska metoder.

Uppgift 3 Använd det ovan för att göra GUI:et lokaliserbart (allt behöver inte fungera bara så att man inser att vi lätt skulle kunna göra klart).

12 Android App

(Frivilligt) Så vitt jag kan bedöma skall vi kunna flytta hela applikationen, förutom GUI:et, till någon Android-baserad telefon. Om du har tid och lust prova...

13 Redovisning

Körningsgodkännande samt kodgenomgång. Görs under laborationspassen. Se till att bli avprickad.

Inlämningsdatum Se kurssida.