Laboration 1 del 2: En översättare

1 Syfte

- Fortsättning på programmering mot gränssnitt och designmönster.
- Förståelse för komponentbaserad programutveckling.
- Konstruktion av applikationer med grafiska användargranssnitt.

2 Uppgift

Vi skall nu använda det vi utvecklade i del 1 (kallas i fortsättningen; modulen) 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.

Figur 1: Engelska till Svenska med svenskt GUI och Engelskt tangentbord.

Figur 2: Svenska till Engelska med Engelskt GUI och Svenskt tangentbord.

3 Kravspecifikation

Detta gäller translator-delen samt kopplingen till den utvecklade modulen.

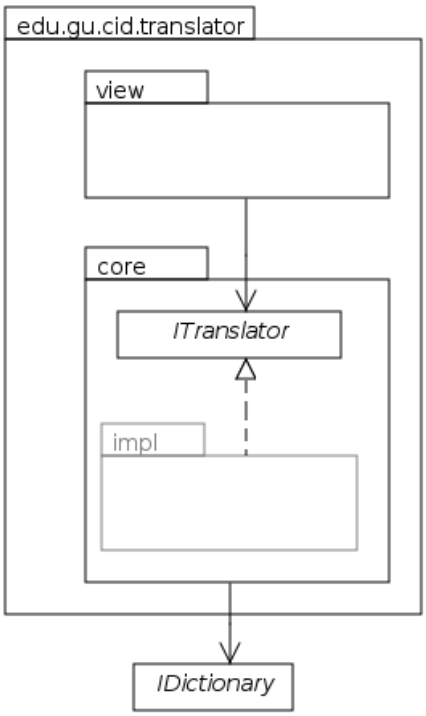
3.1 Funktionalitet

Oförändrad. Finns angiven i del 1.

3.2 Design

3.3 MVC

Applikationen skall byggs enligt MVC-modellen men utan separat C-del efter-



Figur 3: Design. view är vyn, core är modellen, c-del saknas.

som applikationen är så pass enkel. I Figur 3 gör alltså V-delen direkta anrop på M-delens gränssnitt (ITranslator). Klasserna i V-delen har (inre) lyssnare kopplade till GUI-komponenterna. I dessa görs anropen på modellen.

Implementering av Observer mönstret sker m.h.a. färdiga Java klasser och gränssnitt (PropertyChangeSupport, PropertyChangeListener). Alla observerbara klasser implementerar gränssnittet IObserver.

Förändringar i GUI:et kan ske på två sätt;

- I lyssnaren före anropet på modellen (om GUI:et eventuellt ändras av modellanropet)
- I lyssnaren efter anropet på modellen (om vi vet att GUI:et inte kommer att påverkas av anropet).
- Modellen signalerar m.h.a. Observer-mönstret att tillståndet har förändrats. Förändringarna görs då i vyklasssens callback-metod (propertyChange()).

3.4 Människa datorinteraktion

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

Tangentbordet anpassas till utgångsspråket, byter man till svenska skall tangenter för å, ä och ö finnas. In- och utmatningsrutor skall ha "Titled borders" där titeln sätts utifrån modellens tillstånd (är utgångsspråk sv_SV skall detta synas i titelraden), se Figurer.

Menyn File har bara Exit som val och Menyn About visar upp en dialogruta.

3.5 i18n

(BONUSPOÄNG) I i18n-paketet finns en del kod för att göra GUI:et lokaliserbart (byta mellan engelska och svenska), se Figurer. GUI texterna ligger i properties-filer. Se nätet hur man använder dessa tillsammans med klassen ResurceBundle. Språkbyte för GUI görs under menyn Options. Se också klassen Options.

4 Utvecklingsprocess

1. Hämta startkod till labben. Packa upp och importera i Eclipse. En del fel dyker upp.

   Detta projekt är beroende av dictionary-projektet. Markera

translator.ep > Build Path > Configure Build Path ... > Projects > Välj projekt > OK. Felen skall nu ha försvunnit.

2. Vilka metoder skall finnas i ITranslator (vad behöver GUI:et)?

3. Låt klassen Translator (som är en Singleton) implementera gränssnittet, testa metod för metod som tidigare. Troligen behöver ni extra metoder som bara används vid testning (ingår inte ti gränssnittet, vi testar implementationen).

4. Försök nu at koppla ihop GUI:et med modellen. Använd PropertyChanges och StateChanges i Translator och låt modellen signalera vad som hänt. GUI:et fångar händelserna i propertyChange.

5. (BONUSPOÄNG) Gör så att man kan byta språk i GUI:et.

5 Redovisning

Som tidigare.

Inlämningsdatum Se kurssida.