

Lista 7

Uwaga techniczna: Projekt należy skompilować za pomocą “javac lista7/*.java”, a uruchamiać spoza katalogu lista7 za pomocą “java lista7/Main”. Jeśli uruchomi się program będąc wewnątrz katalogu lista7, uzyska się wyjątek “NoClassDefFoundError”.

Event Aggregator:

Implementacja wzorca event aggregator w Javie jest uboższa i zdecydowanie gorsza niż wzorcowa z wykładu w C#. Ponieważ w Javie nie można w trakcie wykonywania programu sprawdzić typu obiektu (Nie ma odpowiednika C#'owego typeof) oraz nie jest możliwa podwójna realizacja generycznego interfejsu (Nie można być Isubscriberem<A> oraz Isubscriberem na raz), zaimplementowałem wzorzec następująco:

- Isubscriber nie jest interfejsem generycznym, klasy realizujące interfejs Isubscriber mają metodę HandleNotifcation(Object notification), w której za pomocą mechanizmu instanceof odsyłają wydarzenie do odpowiedniego handlera.
- Event aggregator nie ma słownika <TypWydarzenia, ListaSubskrybentów>, lecz słownik <String, ListaSubskrybentów>, gdzie stringiem jest kod wydarzenia. Jest to kiepskie rozwiązanie, ponieważ wymaga ustalenia jakichś kodów dla wydarzeń (np. Nazwa klasy).

Wszystkie 4 podpunkty zostały spełnione.

Klasy IeventAggregator, EventAggregator, Isubscriber, są w miarę jasne.

Oprócz nich występują podklasy znanych komponentów Java Swing, które implementują interfejs ISubscriber: MyJTree, KartotekaJPanel, ListaUzytkownikowJPanel.

KartotekaEditDialog, AddUsersDialog to okna modalne umożliwiające to, co ich nazwa wskazuje oraz publikujące eventy do event aggregatora.

KartotekaEditDialogClose oraz AddUsersDialogConfirm to klasy reprezentujące wydarzenia związane z zamknięciem okien modalnych. Zawierają w sobie dane dodanego/zmodyfikowanego użytkownika.

User jest klasą reprezentującą użytkownika, DataBank to klasa gromadząca w sobie użytkowników oraz wydająca dane komponentom Swing za pomocą metod GetUsers(), GetData().

Main inicjalizuje wszystkie JFrame'y etc.