

# Java Swing

## Platformy technologiczne

Michał Piotrowski

8 kwietnia 2010

# Swing

- Swing to biblioteka widgetów do tworzenia okienkowych interfejsów użytkownika w Javie
- Swing jest obiektowo zaprojektowaną biblioteką wykorzystującą wiele typowych obiektowych wzorców projektowych
- Swing pozwala tworzyć interfejsy użytkownika o dowolnym wyglądzie (zgodnym z systemem operacyjnym lub nie)
- By dobrze pisać programy w Swingu należy zapoznać się zarówno z dokumentacją do klas w postaci *javadoc* jak i oficjalnym tutorialiem do Swinga
- Do tworzenia interfejsów użytkownika w Swingu wygodnie jest używać dobrych narzędzi IDE, ale nie jest konieczne (po zrozumieniu koncepcji stosowanych w Swingu pisanie aplikacji korzystających z tej biblioteki nie jest trudne!)

# Obiektowe wzorce projektowe wykorzystywane w Swingu

- Dziedziczenie
- Anonimowe obiekty
- Java Beans
- Obserwator
- Fabryka
- Delegacja
- Model — widok

# Główne klasy komponentów w Swingu

- Kontenery główne (np. okno)
- Kontenery ogólnego zastosowania (np. panel, panel z zakładkami)
- Kontener specjalnego przeznaczenia (np. okno wewnętrzne w interfejsach MDI)
- Podstawowe kontrolki (np. przyciski)
- Edytowalne kontrolki wyświetlające treść w postaci sformatowanej (wybór plików, edytor tekstowy)
- Nieedytowalne kontrolki wyświetlające treść

# Zarządzanie układem komponentów w interfejsie użytkownika

- Do zarządzania układem komponentów w Swingu służą odpowiednie klasy (*layout managers*)
- Zarządcy układu mogą być prości (np. układają komponenty w jednym rzędzie), lub złożeni (np. układają komponenty na podstawie zdefiniowanych zależności przestrzennych między komponentami)

# Wielowątkowość w Swingu

- Wszystkie modyfikacje elementów interfejsu użytkownika (i ich treści) muszą odbywać się w specjalnym wątku: wątku obsługi zdarzeń
- Metody obsługi zdarzeń są automatycznie wywoływane we właściwym wątku
- Wszelkie metody modyfikujące interfejs użytkownika muszą trwać krótko, by go nie blokować
- Długo trwające zadania powinny być wykonywane w innym wątku
- Aby móc modyfikować interfejs użytkownika z poziomu innych wątków należy wykorzystać odpowiednie klasy i metody pomocnicze:
  - SwingWorker
  - SwingUtilities.invokeLater()
  - SwingUtilities.invokeAndWait()
  - Swing Timer

# Model zdarzeń

- Do interakcji z użytkownikiem stosuje się model zdarzeń i nasłuchiwaczy zdarzeń (listeners)
- Klasy nasłuchiwaczy zdarzeń:
  - obsługiwane przez wszystkie komponenty (np. mouse listener, key listener)
  - charakterystyczne dla określonych komponentów (np. action listener)

# Złożone komponenty wyświetlające sformatowane informacje

- Komponenty takie jak tabela czy drzewo opierają się na odpowiednim modelu, który tłumaczy struktury danych przechowywane w pamięci na struktury kompatybilne z danym komponentem
- Za wyświetlanie i edytowanie informacji w złożonych komponentach odpowiadają odpowiednie *renderery* i *edytory*
- Można tworzyć własne *renderery* i *edytory* by wyświetlać dane w dowolny sposób

# Wielojęzykowość

- Wielojęzykowość można osiągnąć stosując pliki *properties* lub klasy dziedziczące po klasie *ResourceBundle*
- Do pobierania zlokalizowanych komunikatów należy wykorzystywać klasę *ResourceBundle*, nie klasę *Properties*
- Klasa *ResourceBundle* automatycznie zarządza cachowaniem zasobów, którymi mogą być zarówno teksty, obrazki i in.

# Biblioteka Beans Binding

- Biblioteka zapewniającą automatyczną synchronizację danych pomiędzy interfejsem użytkownika a danymi zawartymi w obiektach Java Beans
- Java Beans muszą obsługiwać tzw. właściwości wiązane (*bound properties*). Wymagają one generowania zdarzeń informujących o zmianach właściwości w beanie.