

## Program szkolenia:

# Kompleksowe tworzenie aplikacji klasy Desktop z wykorzystaniem SWT i JFace

## Informacje ogólne

<b>Nazwa:</b>	<b>Kompleksowe tworzenie aplikacji klasy Desktop z wykorzystaniem SWT i JFace</b>
<b>Kod:</b>	<b>SWT-JFace</b>
<b>Kategoria:</b>	Technologie Desktop
<b>Grupa docelowa:</b>	programiści, projektanci
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Podczas szkolenia uczestnicy posiadają umiejętność posługiwania się Biblioteką SWT i frameworkiem JFace jak również narzędziami pozwalającymi na efektywne tworzenie czytelnego kodu.

Podczas ćwiczeń rozwijamy praktycznych umiejętności tworzenia interfejsu użytkownika, testowania i separacji kodu.

Część teoretyczna daje wyczerpujący przegląd możliwości dostarczanych przez biblioteki i technik ich wykorzystania (data binding, MVC za pomocą JFace viewers), ze szczególnym uwzględnieniem rozwiązań często pojawiających się problemów (zwalnianie zasobów, synchronizacja z wątkiem graficznym).

### Zalety szkolenia:

- » Realne przykłady i najlepsze praktyki
- » Zagadnienia testowania i separacji odpowiedzialności poprzez zastosowanie MVC
- » Typowe pułapki i triki
- » Zaawansowane mechanizmy binding

## Program szkolenia:

### 1. Przegląd architektur aplikacji Desktop

#### 1.1. Model View Controller

##### 1.1.1. Separacja odpowiedzialności kodu

##### 1.1.2. Projektowanie łatwo utrzymywanych aplikacji

#### 1.2. Architektura Warstwowa

##### 1.2.1. Warstwy to nie MVC

#### 1.3. Architektura Eventually Connected Client

#### 1.4. Komunikacja aplikacji z Serverem (EJB/Spring)

#### 1.5. Projektowanie aplikacji otwartej na testy

##### 1.5.1. Aspekty tworzenia testowalnego kodu

##### 1.5.2. Techniki Mockowania Servera

### 2. Tworzenie interfejsu użytkownika z wykorzystaniem SWT, JFace i WindowBuilderPro

#### 2.1. Biblioteka SWT jako połączenie zalet Swing i AWT

#### 2.2. Rozszerzenia JFace do biblioteki SWT

#### 2.3. Efektywna praca z kodem za pomocą WindowBuilderPro (WBPro)

### 3. Komponenty i zdarzenia

#### 3.1. Podstawowe komponenty SWT i techniki ich rozmieszczania

#### 3.2. Model zdarzeniowy SWT

#### 3.3. Cechy komponentów i ich dodatkowe style

#### 3.4. Rozmieszczanie komponentów z wykorzystaniem WBPro

#### 3.5. Tworzenie formularzy z wykorzystaniem najczęściej używanych komponentów

### 4. Zarządcy układu

#### 4.1. Zastosowania zarządców układu i ich charakterystyka

4.2. Właściwości zarządzców rozkładu

4.3. Klasyczne problemy rozmieszczania komponentów

4.3.1. Rozszerzanie na cały ekran

4.3.2. Automatyczne dostosowanie ilości komponentów do długości wiersza/kolumny

4.3.3. Rozmieszczanie komponentów w siatce

4.3.4. Definiowanie położenia w procentowo określonych fragmentach ekranu

4.4. Dostosowywanie szczegółów ułożenia komponentów za pomocą danych układu

4.5. Projektowanie UI z wykorzystaniem kontenerów i zarządzców układu

4.6. Dynamiczne modyfikowanie zawartości ekranu

4.7. Tabele i zarządzcy układu związani z tabelami

4.8. Wsparcie dla zarządzców układu w WBPro

## 5. JFace Viewers

5.1. Wprowadzenie do rozdzielania widoku od modelu za pomocą viewer framework

5.2. Implementacja podstawowych viewerów - dla listy i listy rozwijanej

5.3. Usuwanie elementów i odświeżanie widoku

5.4. Wygodna obsługa tabel z wykorzystaniem TableViewer

## 6. Okna dialogowe

6.1. Dostosowywanie okien komunikatów w SWT za pomocą stylów i ikon

6.2. Standardowe okna dialogowe JFace

6.3. Dedykowane okna dialogowe do wczytywania danych użytkownika i obsługi systemu plików

6.4. Walidacja w oknach dialogowych

6.5. Tworzenie własnych okien dialogowych w oparciu o Shell

## 7. Tworzenie intuicyjnego UI z wykorzystaniem kreatorów

7.1. Organizowanie UI za pomocą kreatorów

7.2. Główna klasa kreatora, tworzenie stron i uruchamianie

7.3. Implementacja akcji na zakończenie kreatora i anulowanie kreatora

7.4. zaawansowana możliwości kreatorów

7.4.1. Blokowanie dostępu do kolejnych stron i zakończenia

7.4.2. Walidacja danych za pomocą mechanizmu komunikatów kreatora

7.4.3. Dynamiczne modyfikowanie kolejności stron kreatora

## 8. Definiowanie menu w SWT

8.1. Organizowanie zadań w aplikacji za pomocą menu głównego

8.2. Dostarczanie pomocniczych zadań z użyciem menu kontekstowego

8.3. Niestandardowe pozycje menu i zagnieżdżanie menu

## 9. Kolory, czcionki, ikony i zwalnianie zasobów

9.1. Wsparcie dla edycji wyglądu graficznego aplikacji

9.2. Zwalnianie zasobów

9.2.1. Monitorowanie wycieków zasobów systemowych w SWT

9.2.2. Implementacja zwalniania zasobów w SWT i JFace

9.2.3. Zwalnianie zasobów za pomocą rejestrów

9.3. Obsługa kolorów, czcionek i ikon w SWT

9.4. Posługiwanie się lekkimi klasami definicji zasobów i oknami wyboru

9.5. Problemy z modyfikacją grafiki w aplikacjach opartych o SWT

## 10. Bindowanie w SWT i JFace

10.1. Mechanizm JFace Data Binding: realm, context, binding dla komponentów graficznych i POJO

10.2. Wsparcie dla bindowania w WBPro

10.3. Bindowanie dwukierunkowe

10.4. Walidacja z wykorzystaniem bindowania

10.5. Modyfikacja procesu bindowania za pomocą walidatorów i konwerterów

10.6. Bindowanie dla JFace viewers

## 11. Wielowątkowość w SWT

11.1. Konsekwencje sposobu implementacji modelu zdarzeniowego w SWT

11.2. Uruchamianie zadań w synchronizacji z wątkiem graficznym

11.3. Komponenty do prezentacji postępu zadania

11.4. Uruchamianie zadań asynchronicznie i z opóźnieniem

## 12. Tworzenie własnych komponentów

12.1. Ograniczenia związane z dziedziczeniem w SWT

12.2. Typowe elementy komponentów, wymagające specjalnej implementacji

## 13. Pozostałe elementy SWT i JFace

13.1. Upraszczenie złożonych interfejsów użytkownika z wykorzystaniem zakładek, rozwijanych sekcji

13.2. Organizacja okna aplikacji z wykorzystaniem zakładek, pasków narzędziowych i obszarów ekranu pozwalających na zmianę rozmiaru

13.3. Organizacja menu, paska stanu, pasków narzędziowych za pomocą JFace Application Window

13.4. Wprowadzenie do EclipseRCP