

# Eclipse

M. Albrycht   Ł. Zubkowicz

24 marca 2009

# Ogólnie o Eclipse

- framework dla IDE (ang. Integrated Development Environment)
- rozszerzalny przez wtyczki (bez nich nie ma mowy o kompletnym IDE)
- wtyczki wspierające nie tylko Javę (np. C++, PHP)

# Ogólnie o Eclipse

- napisany w Javie (dzięki temu przenośny)
- interfejs stworzony w oparciu o SWT (specjalnie stworzone w tym celu)
- projekt zapoczątkowany przez IBM w 2001 roku
- tworzony przez społeczność wspieraną przez “The Eclipse Foundation”
- darmowy, open-source (licencja EPL - Eclipse Public License)

# Platforma Eclipse

- podstawa wszystkich innych funkcjonalności (to ją trzeba zainstalować)
- udostępnia wewnętrzny instalator przez który pobieramy i instalujemy wtyczki
- posiada wbudowany mechanizm automatycznej aktualizacji wtyczek

## Funkcje wtyczek

- wspieranie konkretnych języków (np. JDT - Java, PDT - PHP, CDT - C++)
- dodawanie narzędzi do modelowania i projektowania (np. MDT)
- automatyzacja wyrafinowanych procesów budowania aplikacji
- wspomaganie testów jednostkowych i wydajnościowych
- dodawanie obsługi repozytoriów danych (np. Subclipse - SVN)
- konfiguruja i zespala narzędzi dla najróżniejszych platform (np. MTJ - Java ME, JEE Tools - Java EE)

# Główne okno aplikacji

- menu i pasek narzędzi
- składa się z edytora i tzw. widoków (ang. views) jak np. Navigator, Outline, Hierarchy
- zbiory widoków to perspektywy (służące do zmiany wielu widoków na raz)



## Funkcjonalność edytora

- sprawdzanie pisowni, dokańczanie słów
- domykanie np. nawiasów i cudzysłówów
- wstawianie gotowych szablonów konstrukcji językowych
- podpowiadanie rozwiązań dla błędów (QuickFix)
- ustawianie zakładek w plikach
- i wiele innych

## Widoki i perspektywy

- różne strukturalne informacje i opcje podzielone na okienka zwane widokami
- ogromne możliwości konfiguracji wzajemnego ułożenia widoków
- rozkład swojej przestrzeni pracy można zapisać jako perspektywa (istnieje wiele domyślnych perspektyw)



## Ogólnie

- przystosowuje mechanizmy środowiska do tworzenia w języku Java
- dodaje nowe perspektywy (np. Java)
- pozwala tworzyć projekty javowe
- udostępnia wiele kreatorów specyficznych dla wspomnianych projektów
- “uczy” środowisko asystować przy tworzeniu oprogramowania
- pozwala na specyficzny dla języka refactoring kodu

## Uruchamianie i debuggowanie

- inkrementacyjna kompilacja kodu (podczas zapisu źródeł)
- zgodność z wieloma wirtualnymi maszynami
- uruchamianie kodu w trybie zwykłym oraz debug
- ponowne ładowanie klas “w locie” (opcja zależna od VM)

## Tworzenie kodu

- kolorowanie składni
- asystowanie przy pisaniu kodu i dokumentacji
- wyświetlanie pomocy i podpowiedzi
- automatyczne formatowanie kodu
- śledzenie nieużywanych zmiennych
- automatyczne dodawanie importów

## Najważniejsze widoki

- Package Explorer - ukazuje drzewo całego projektu, od pakietów aż po klasy
- Outline - w czytelny sposób wyświetla strukturę (streszczenie) otwartego w edytorze pliku
- Hierarchy - przedstawia hierarchię klas danego obiektu (można przeglądać także pola i metody)

# Konfiguracja

- konfiguracja środowiska jest dostępna z menu *Window, Preferences*
- bogactwo opcji konfiguracyjnych często okazuje się zgubne, gdy potrzebujemy zmienić jedną małą, specyficzną rzecz

## Wybrane ustawienia

- czcionki edytora można zmieniać w zakładce *General*, *Appearance*, *Colors and Fonts*
- wsparcie dla polskiego słownictwa (np. gdy ktoś lubi pisać komentarze i dokumentację po polsku) można sobie zapewnić ustawiając polski słownik w zakładce *General*, *Editors*, *Text Editors*, *Spelling*
- polski słownik można ściągnąć np. ze strony `'http://www.sjp.pl/slownik/growy'` (albo można zrobić sobie samemu ;) - jest to plik tekstowy posiadający po jednym wyrazie w linii)

## Bałagan z powiadomieniami o błędach

- inkrementacyjny charakter kompilacji powoduje powstawanie czasem różnych niespójności
- objawia się to w ten sposób, że w pewnym momencie mamy dużo błędów i nie wiemy skąd pochodzą, albo wydają się bez sensu
- rozwiązaniem najczęściej jest wyczyszczenie projektu i zbudowanie go od nowa korzystając z menu *Project, Clean...*

## Skróty klawiszowe bardzo przyspieszają pracę

- standardowe: CTRL+S, CTRL+X, CTRL+C, CTRL+V, CTRL+Z, CTRL+Y (Redo)
- lista otwartych pliki: CTRL+E, zamknięcie pliku: CTRL+W, maksymalizacja okna edytora: CTRL+M (powrót tak samo)
- zakomentowanie/odkomentowanie zaznaczonych linii: CTRL+SHIFT+C, przeniesienie linii wyżej/niżej: ALT+(UP/DOWN), usunięcie linii: CTRL+D



## Plan części pokazowo-ćwiczeniowej

- Początek pracy, ustawienie Workspace'u
- Importowanie projektu
- Organizacja pracy z projektami
- Własny projekt
- Omówienie okien programu
- Template
- Debugger
- Export projektu

## Potrzebne linki

### Pobieranie Eclipse:

Moja strona (Windows)	<a href="http://students.mimuw.edu.pl/~ma248357/eclipse/">http://students.mimuw.edu.pl/~ma248357/eclipse/</a>
Oficjalna strona	<a href="http://www.eclipse.org/">http://www.eclipse.org/</a>

### Repozytorium, z którego będziemy korzystać:

[http://subversion.assembla.com/svn/TEST\\_theo1](http://subversion.assembla.com/svn/TEST_theo1)

### URL potrzebny do instalacji subclipse'a:

<http://subclipse.tigris.org/update-1.4.x>

### Lista Warningów:

[http://jazzy.id.au/pebble/2008/10/30/list\\_of\\_suppresswarnings\\_arguments.html](http://jazzy.id.au/pebble/2008/10/30/list_of_suppresswarnings_arguments.html)

## Bibliografia

- <http://www.ibm.com/developerworks/opensource/library/os-eclipse-platform/>
- <http://pl.wikipedia.org/wiki/Eclipse>
- <http://eclipsutorial.sourceforge.net>
- <http://www.eclipse.org/>