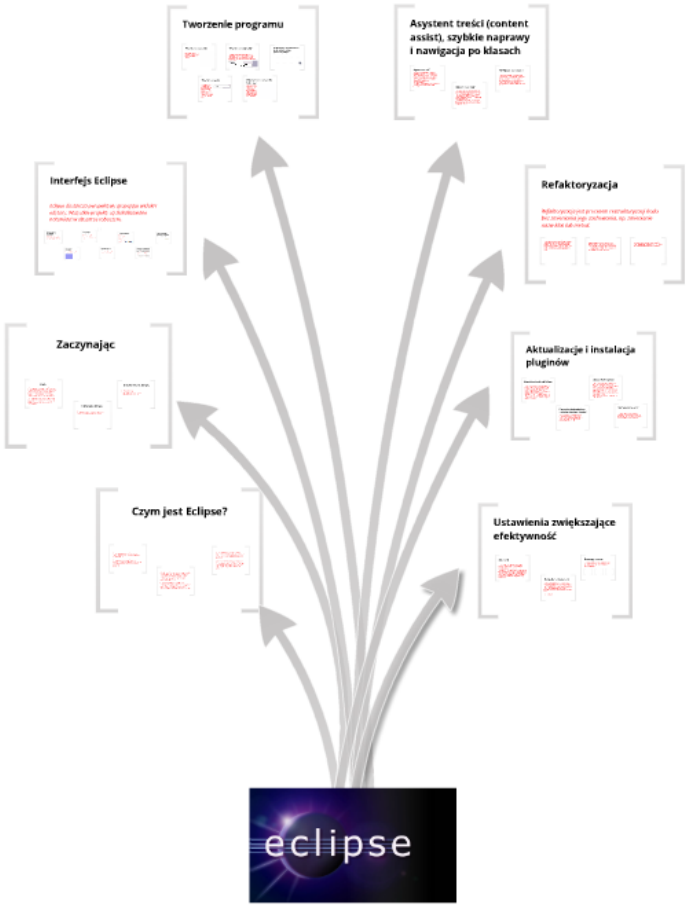


Dziękuję za uwagę :)


eclipse

Czym jest Eclipse?

Eclipse został stworzony przez firmę IBM, a następnie udostępniony na zasadach otwartego oprogramowania.

Eclipse w założeniach miał być rodzajem uniwersalnej platformy – otwartego, rozszerzalnego IDE (Integrated Development Environment) do wszystkiego.

Szczególną siłę Eclipse daje mechanizm wtyczek, który jest podstawą działania środowiska. Umożliwia on pracę nad różnego rodzaju projektami, nie tylko programistycznymi. Dużą rolę także stanowi wieloplatformowość Eclipse.

Większość ludzi uważa Eclipse głównie jako zintegrowane środowisko deweloperskie dla Javy. W tej kwestii faktycznie sprawuje się bardzo dobrze zgarniając 65% rynku środowisk developerskich „Javy".

Eclipse jest tworzony przez społeczność Open Source i dzięki temu jest używany w kilku różnych dziedzinach, np. jako środowisko Javy dla aplikacji Androida.

Obecnie Eclipse jest zarządzany przez Fundację Eclipse. Jest to fundacja non-profit a jej członkowie pomagają tworzyć ekosystem społeczności open source zapewniający rozwój produktu i jego wsparcie.

Eclipse został stworzony przez firmę IBM, a następnie udostępniony na zasadach otwartego oprogramowania.

Eclipse w założeniach miał być rodzajem uniwersalnej platformy – otwartego, rozszerzalnego IDE (Integrated Development Environment) do wszystkiego.

Szczególną siłę Eclipse daje mechanizm wtyczek, który jest podstawą działania środowiska. Umożliwia on pracę nad różnego rodzaju projektami, nie tylko programistycznymi. Dużą rolę także stanowi wieloplatformowość Eclipse.

Większość ludzi uważa Eclipse głównie jako zintegrowane środowisko deweloperskie dla Javy. W tej kwestii faktycznie sprawuje się bardzo dobrze zgarniając 65% rynku środowisk developerskich „Javy".

Eclipse jest tworzony przez społeczność Open Source i dzięki temu jest używany w kilku różnych dziedzinach, np. jako środowisko Javy dla aplikacji Androida.

Obecnie Eclipse jest zarządzany przez Fundację Eclipse. Jest to fundacja non-profit a jej członkowie pomagają tworzyć ekosystem społeczności open source zapewniający rozwój produktu i jego wsparcie.

Zaczynając

Java

Eclipse wymaga zainstalowanej Javy. Tą możemy spotkać jako Java Runtime Environment (JRE) oraz Java Development Kit (JDK). JRE umożliwia tylko używanie programów w Javie podczas gdy JDK zawiera dodatki potrzebne do programowania.

JDK jest wymagane podczas kompilowania kodu Javy poza Eclipse oraz bardziej specjalistycznego używania javy, jak podczas pisania aplikacji internetowych.

Instalacja Eclipse

Ściągamy ze strony eclipse.org interesującą nas pozycję i rozpakowujemy. Ot cała filozofia :)

Uruchomienie Eclipse

Eclipse uruchamiamy

- *w Windowsie poprzez eclipse.exe*
- *w Linux/Mac poprzez eclipse*

Java

Eclipse wymaga zainstalowanej Javy. Tą możemy spotkać jako Java Runtime Environment (JRE) oraz Java Development Kit (JDK). JRE umożliwia tylko używanie programów w Javie podczas gdy JDK zawiera dodatki potrzebne do programowania.

JDK jest wymagane podczas kompilowania kodu Javy poza Eclipse oraz bardziej specjalistycznego używania javy, jak podczas pisania aplikacji internetowych.

Instalacja Eclipse

Ściągamy ze strony eclipse.org interesującą nas pozycję i rozpakowujemy. Ot cała filozofia :)

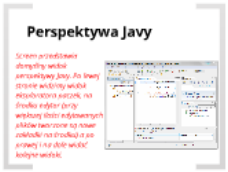
Uruchomienie Eclipse

Eclipse uruchamiamy

- *w Windowsie poprzez eclipse.exe*
- *w Linux/Mac poprzez eclipse*

Interfejs Eclipse

Eclipse dostarcza perspektywy grupujące widoki i edytory. Wszystkie projekty są zlokalizowane natomiast w obszarze roboczym.

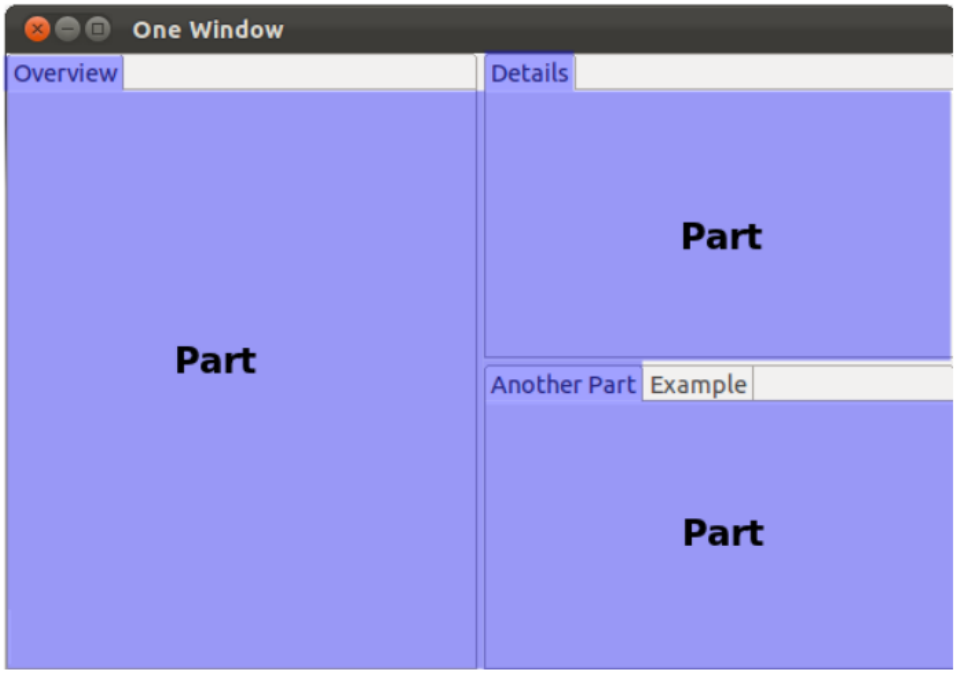


Obszar roboczy (workspace)

Jest miejscem fizycznego przechowywania plików, na których się pracuje. Obszar roboczy można wybrać podczas uruchomienia Eclipse albo poprzez menu (File Switch Workspace Others). Wszystkie projekty, pliki źródłowe, obrazki i inne będą przechowywane i zapisywane w tym miejscu.

Karty (parts)

Są to komponenty interfejsu użytkownika, które pozwalają na nawigowanie oraz modyfikowanie danych. Karty są podzielone na widoki i edytory.



Różnice między widokami a edytorami nie są oparte na technicznych różnicach lecz na koncepcji ich używania.

Widok jest zazwyczaj używany do pracy na zestawach danych, które mogą być uporządkowane hierarchicznie. Jeżeli dane są zmienione za pomocą widoku to zmiana ta zwykle odnosi się do podstawowej struktury danych (np. po zmianie nazwy pliku ów plik zmieni swoją nazwę również w systemie). Widok czasami umożliwia otwarcie Edytora dla wybranego zestawu danych.

Edytory natomiast zwykle są używane do modyfikowania pojedynczych elementów z danymi, np. plików. Tutaj aby zatwierdzić zmiany użytkownik musi zapisać plik nowymi danymi z okna edytora.

Edytory tradycyjnie umieszczone są w centralnym obszarze środowiska. Do Eclipse 4 nie można było przenieść edytora poza ten obszar jednak od Eclipse 4 umożliwiono umieszczenie edytora w dowolnym miejscu w perspektywie.

Różnice między widokami a edytorami nie są oparte na technicznych różnicach lecz na koncepcji ich używania.

Widok jest zazwyczaj używany do pracy na zestawach danych, które mogą być uporządkowane hierarchicznie. Jeżeli dane są zmienione za pomocą widoku to zmiana ta zwykle odnosi się do podstawowej struktury danych (np. po zmianie nazwy pliku ów plik zmieni swoją nazwę również w systemie). Widok czasami umożliwia otwarcie Edytora dla wybranego zestawu danych.

Edytory natomiast zwykle są używane do modyfikowania pojedynczych elementów z danymi, np. plików. Tutaj aby zatwierdzić zmiany użytkownik musi zapisać plik nowymi danymi z okna edytora.

Edytory tradycyjnie umieszczone są w centralnym obszarze środowiska. Do Eclipse 4 nie można było przenieść edytora poza ten obszar jednak od Eclipse 4 umożliwiono umieszczenie edytora w dowolnym miejscu w perspektywie.

Perspektywa

Perspektywa jest wizualnym kontenerem na karty. Środowisko Eclipse używa perspektyw do organizowania kart dla poszczególnych programistycznych zadań. Nie są to jednak sztywne ustawienia i można dostosowywać również wygląd otwierając bądź zamykając karty wg własnych upodobań.

Perspektywy w Eclipse

Pisząc w Javie zazwyczaj używa się perspektywy Javy, lecz Eclipse posiada wiele więcej zdefiniowanych perspektyw, np. debugowanie, repozytoria GIT i CVS.

Zmienić perspektywę można za pomocą menu Window Open Perspective Other.

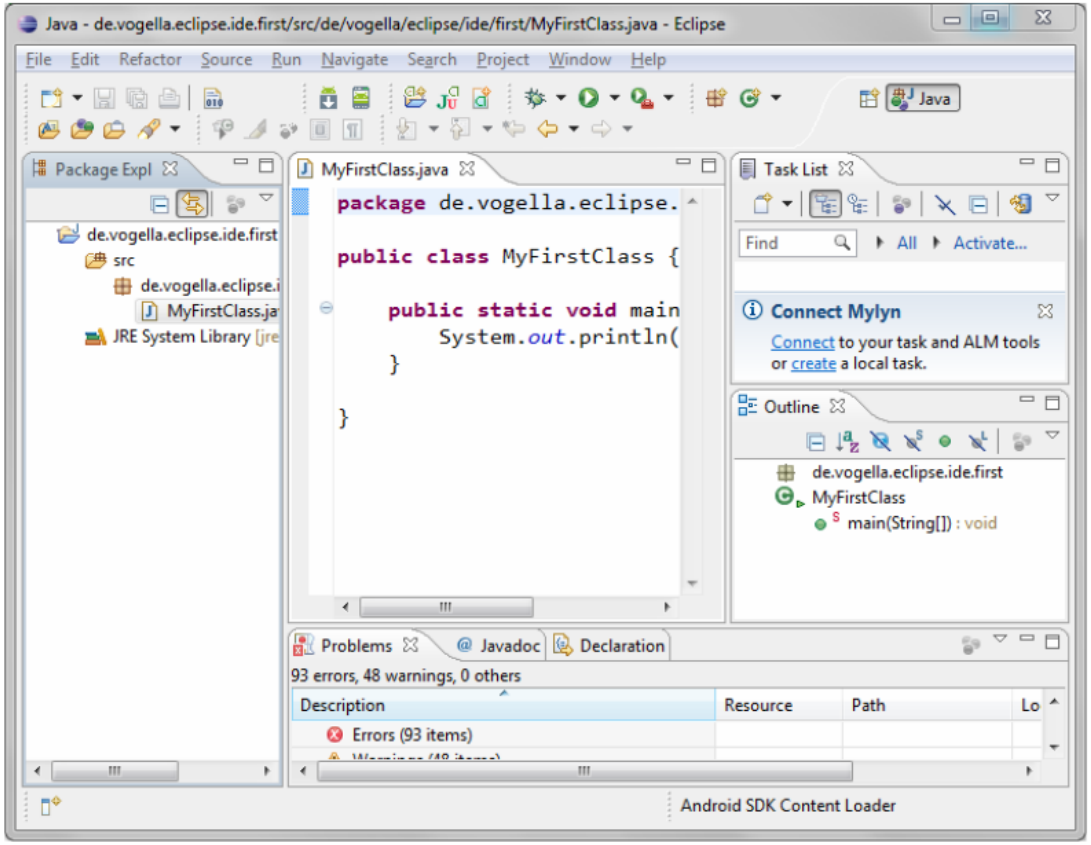
Jeżeli chcemy przywrócić domyślny widok perspektywy można ją zresetować używając menu Window Reset Perspective.

Eksplorator paczek

Eksplorator paczek znajduje się zazwyczaj po lewej stronie. Umożliwia on na przeglądanie projektów i zaznaczanie komponentów z którymi chcemy pracować.

Perspektywa Javy

Screen przedstawia domyślny widok perspektywy Javy. Po lewej stronie widzimy widok eksploratora paczek, na środku edytor (przy większej ilości edytowanych plików tworzone są nowe zakładki na środku) a po prawej i na dole widać kolejne widoki.

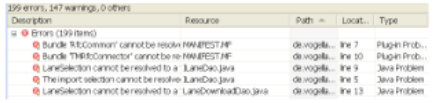


Połączenie eksploratora paczek z edytorem kodu

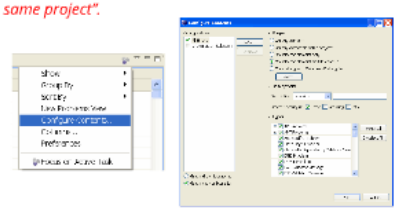
Eksplorator paczek umożliwia wyświetlanie powiązanych plików z aktualnie edytowanym plikiem. Aby aktywować tą opcję należy użyć przycisku „Link with Editor" w widoku eksploratora paczek.

Okno wyświetlające problemy z programem

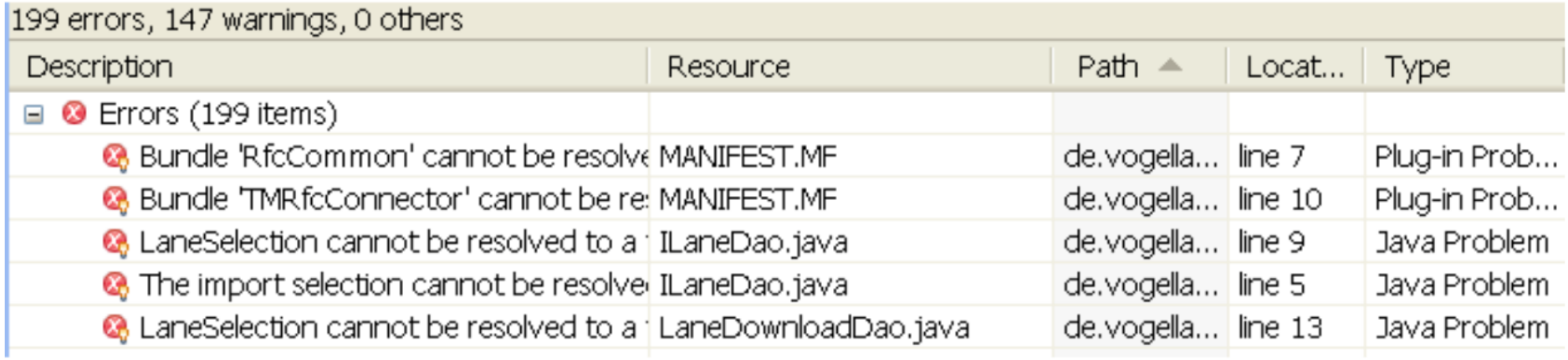
Prędzej czy później każdy spotyka się z problemami w kodzie bądź też ustawień projektu. Aby zobaczyć co je spowodowało należy użyć widoku „Problems" widocznego zazwyczaj na dole.



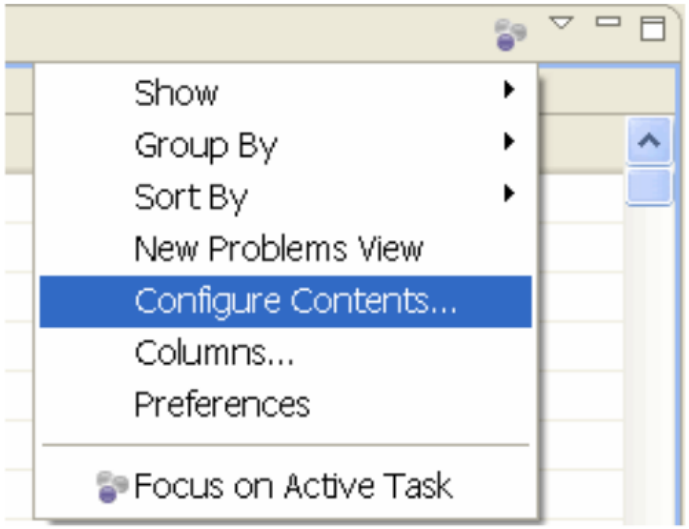
Można również edytować ilość zawartych informacji o błędach tak, aby na przykład wyświetlać błędy dla całego projektu ustawiamy „On any element in the same project".

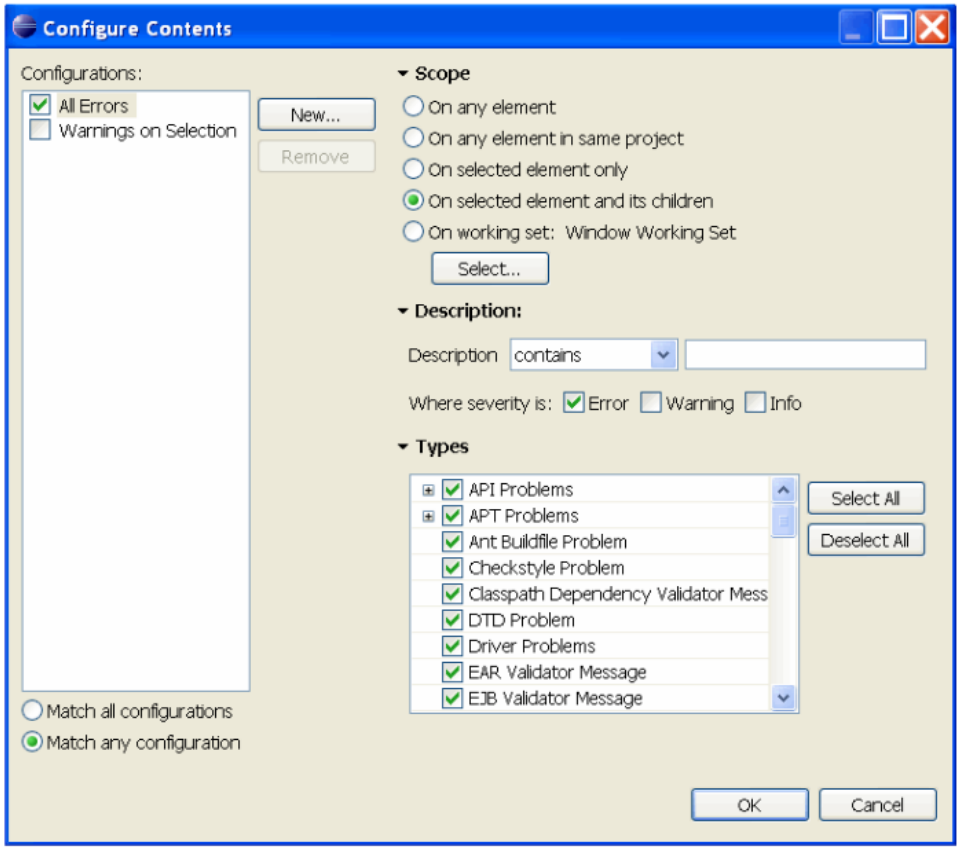


Prędzej czy później każdy spotyka się z problemami w kodzie bądź też ustawień projektu. Aby zobaczyć co je spowodowało należy użyć widoku „Problems" widocznego zazwyczaj na dole.



Można również edytować ilość zawartych informacji o błędach tak, aby na przykład wyświetlać błędy dla całego projektu ustawiamy „On any element in the same project”.





Tworzenie programu

Tworzenie projektu

Wybieramy z menu File New Java project. W polu Project name: wpisujemy nazwę aplikacji.

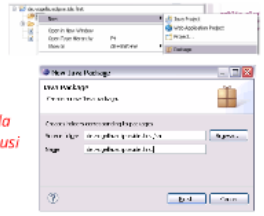
Tworzenie klasy javy

Klikamy prawym przyciskiem myszy na paczkę i wybieramy New Class. Tworząc główną klasę poza wpisaniem nazwy zaznaczamy „public static void main(String[] args) aby utworzyło nam automatycznie kawałek kodu.

Przygotowanie do uruchomienia programu poza Eclipse (tworzenie pliku jar)

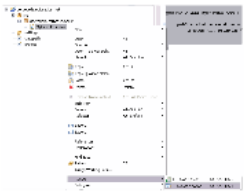
Tworzenie paczki

Szukamy folder src, klikamy na niego prawym przyciskiem myszy wybieramy z menu New Package. Wpisujemy nazwę (dla Javy nazwa paczki musi składać się z małych liter).



Uruchomienie projektu w Eclipse

Wystarczy kliknąć na wybranej przez nas klasie prawym przyciskiem myszy i wybrać Run As Java Application. Wszystko powinno uruchomić się bez problemów.

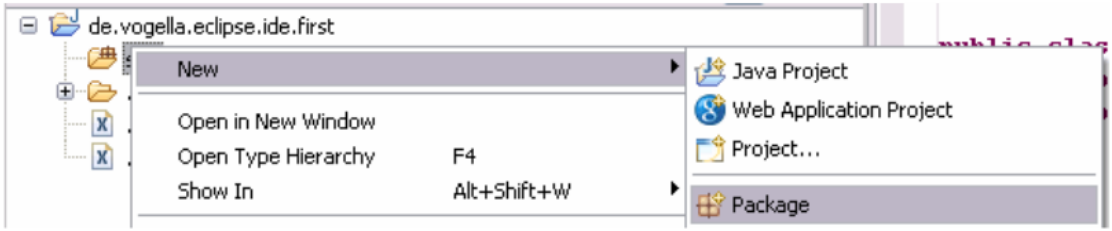


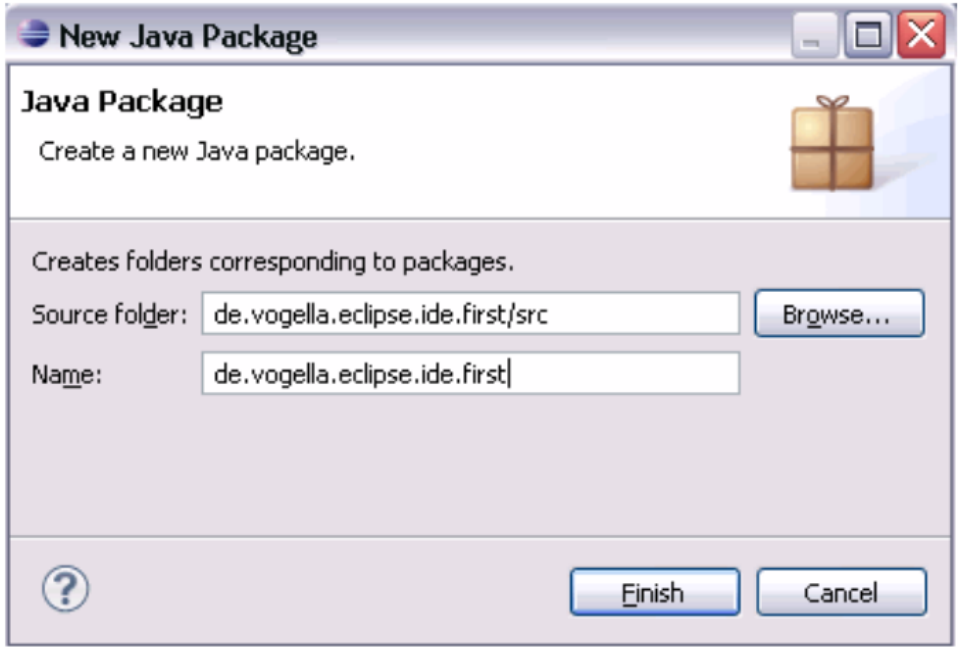
Tworzenie projektu

Wybieramy z menu File New Java project. W polu Project name: wpisujemy nazwę aplikacji.

Tworzenie paczki

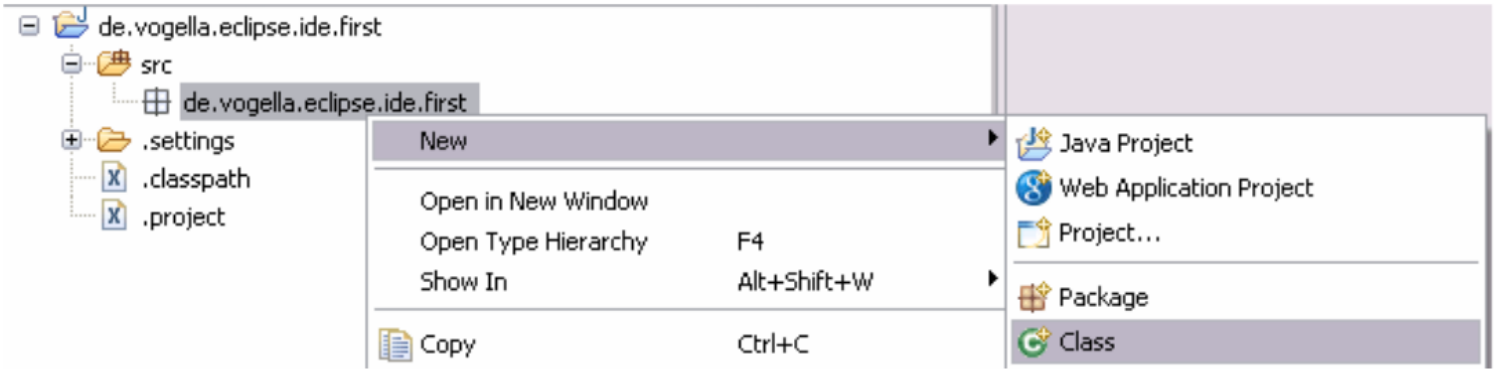
Szukamy folder src, klikamy na niego prawym przyciskiem myszy wybieramy z menu New Package. Wpisujemy nazwę (dla Javy nazwa paczki musi składać się z małych liter).

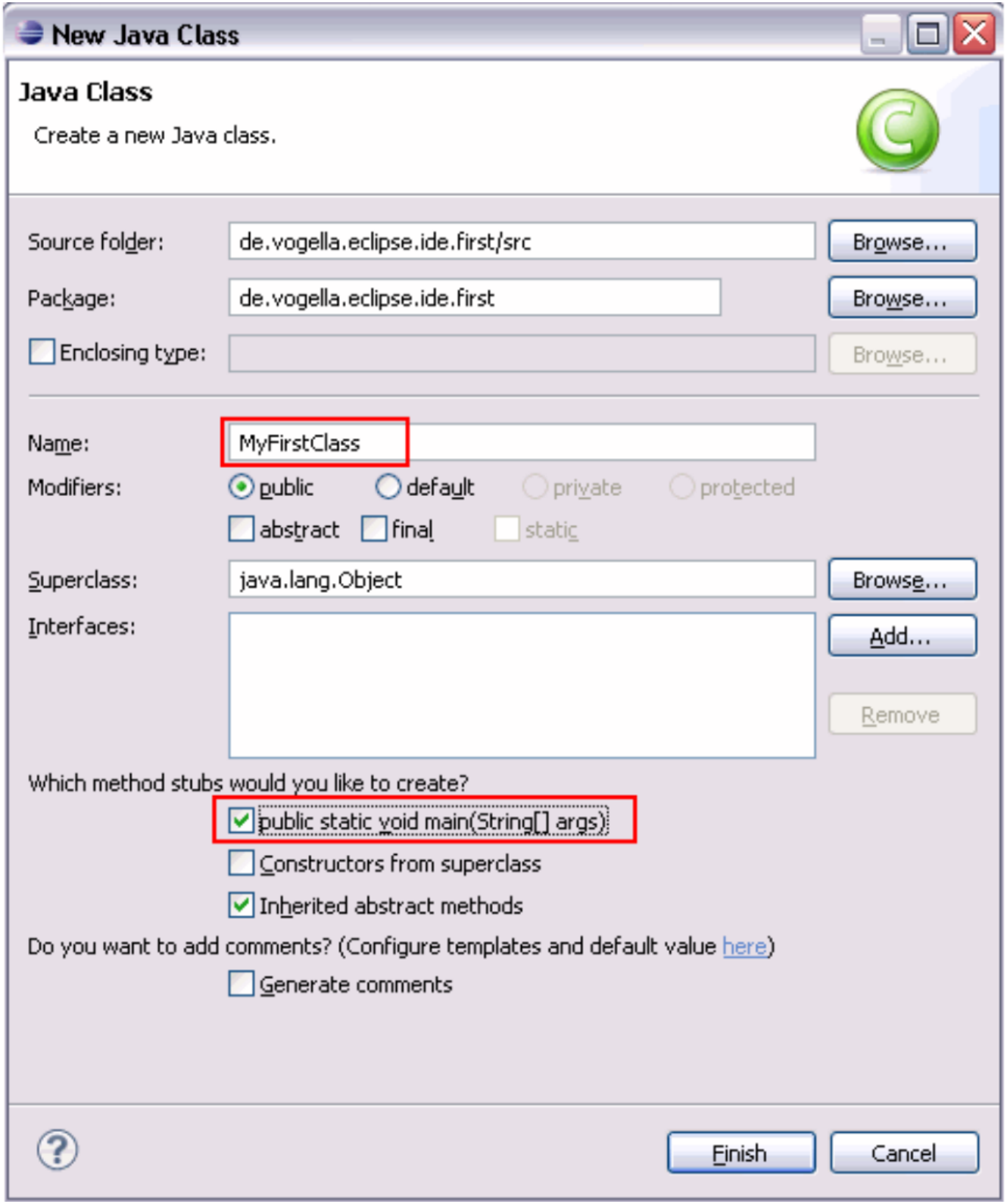




Tworzenie klasy javy

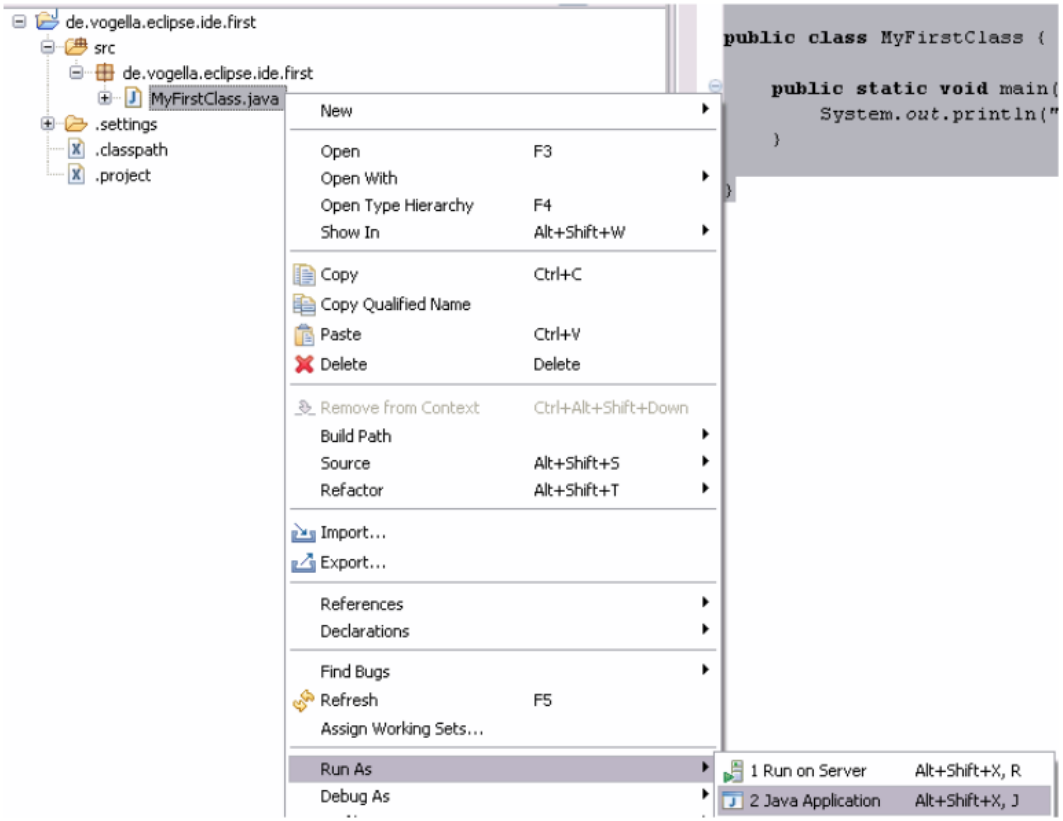
Klikamy prawym przyciskiem myszy na paczkę i wybieramy New Class. Tworząc główną klasę poza wpisaniem nazwy zaznaczamy „public static void main(String[] args) aby utworzyło nam automatycznie kawałek kodu.




New Java Class
Java Class
Create a new Java class.
Source folder:
de.vogella.eclipse.ide.first/src
Browse...
Package:
de.vogella.eclipse.ide.first
Browse...
Enclosing type:
Browse...
Name:
MyFirstClass
Modifiers:
public
default
private
protected
abstract
final
static
Superclass:
java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Finish
Cancel

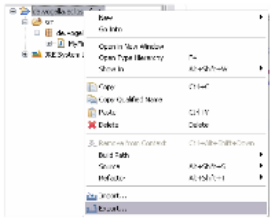
Uruchomienie projektu w Eclipse

Wystarczy kliknąć na wybranej przez nas klasie prawym przyciskiem myszy i wybrać Run As Java Application. Wszystko powinno uruchomić się bez problemów.

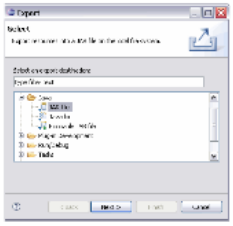


Przygotowanie do uruchomienia programu poza Eclipse (tworzenie pliku jar)

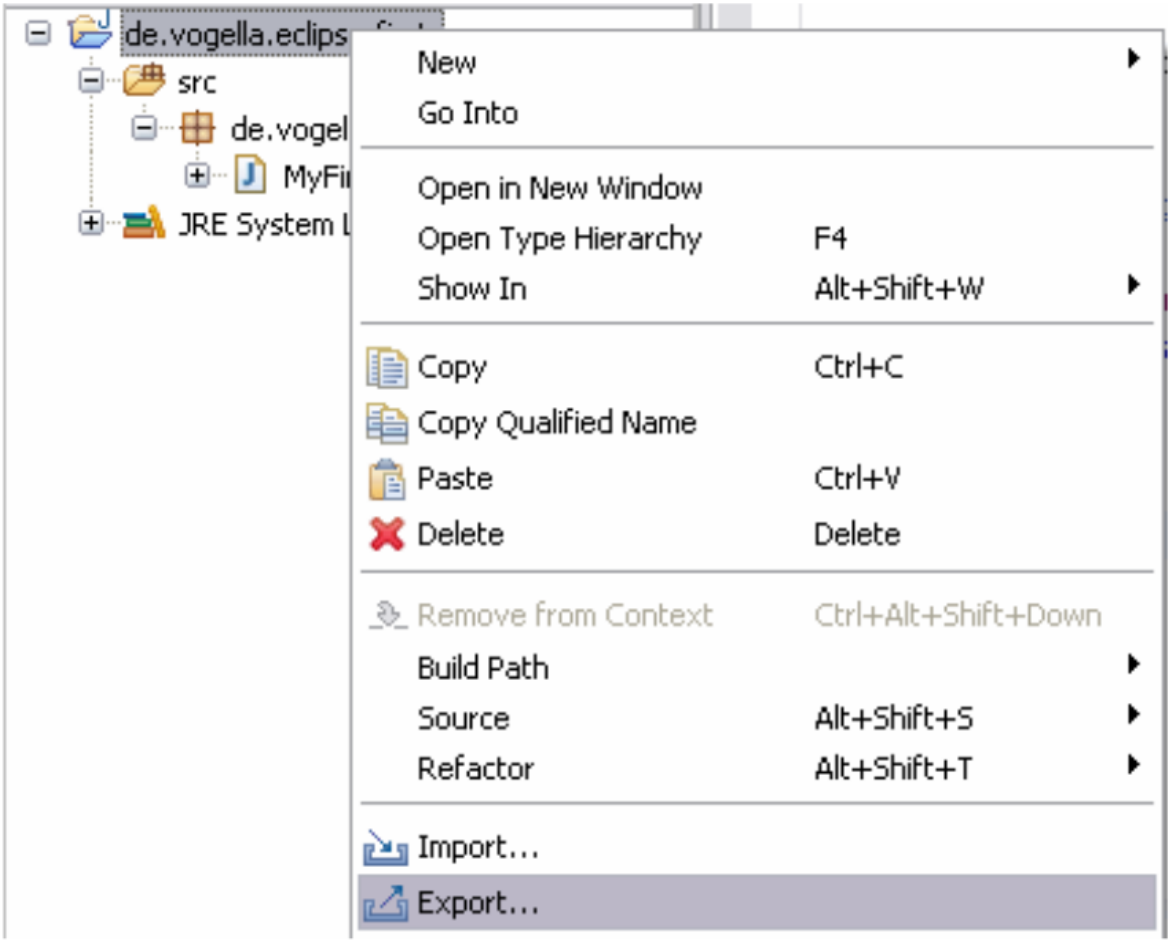
Aby uruchomić program poza środowiskiem Javy należy przygotować plik .jar (jest to domyślne rozszerzenie dla aplikacji Javy). Klikamy na projekt prawym przyciskiem myszy i wybieramy Export.



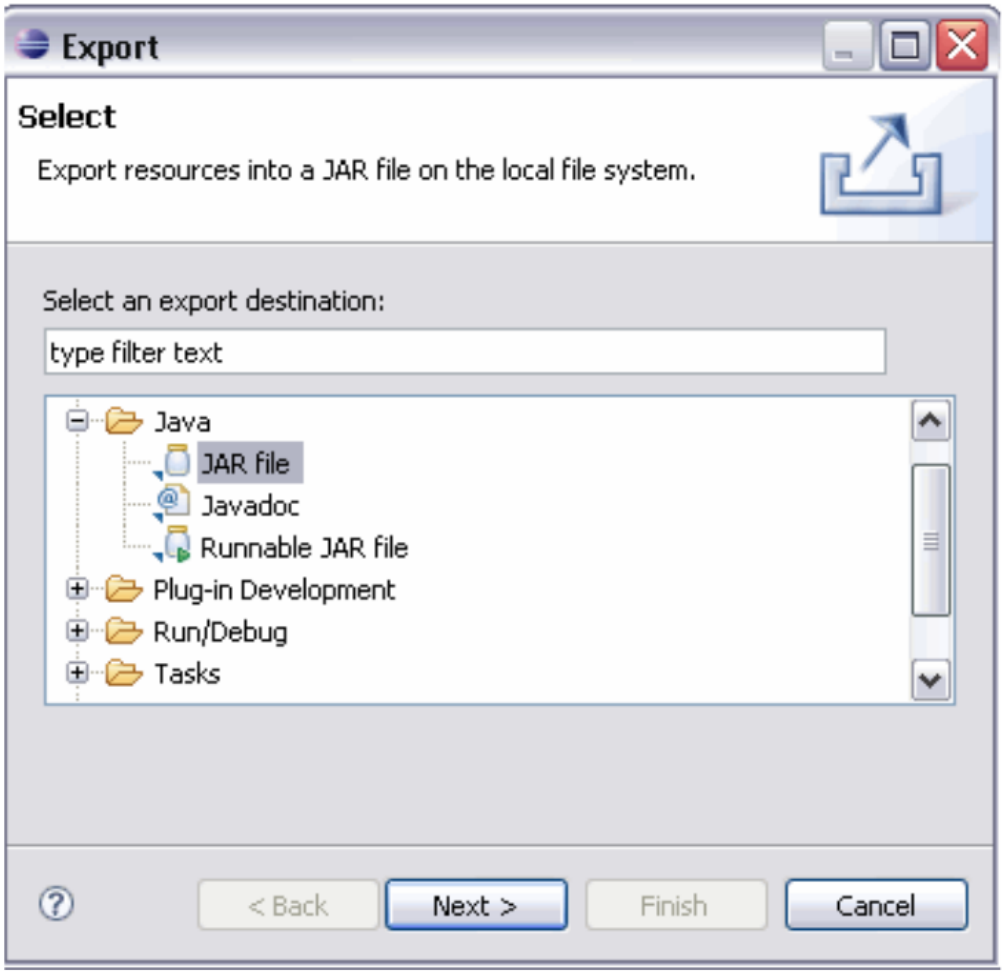
Wybieramy Java JAR file, next po czym wpisujemy ścieżkę zapisu i nazwę pliku .jar i kończymy za pomocą przycisku Finish.

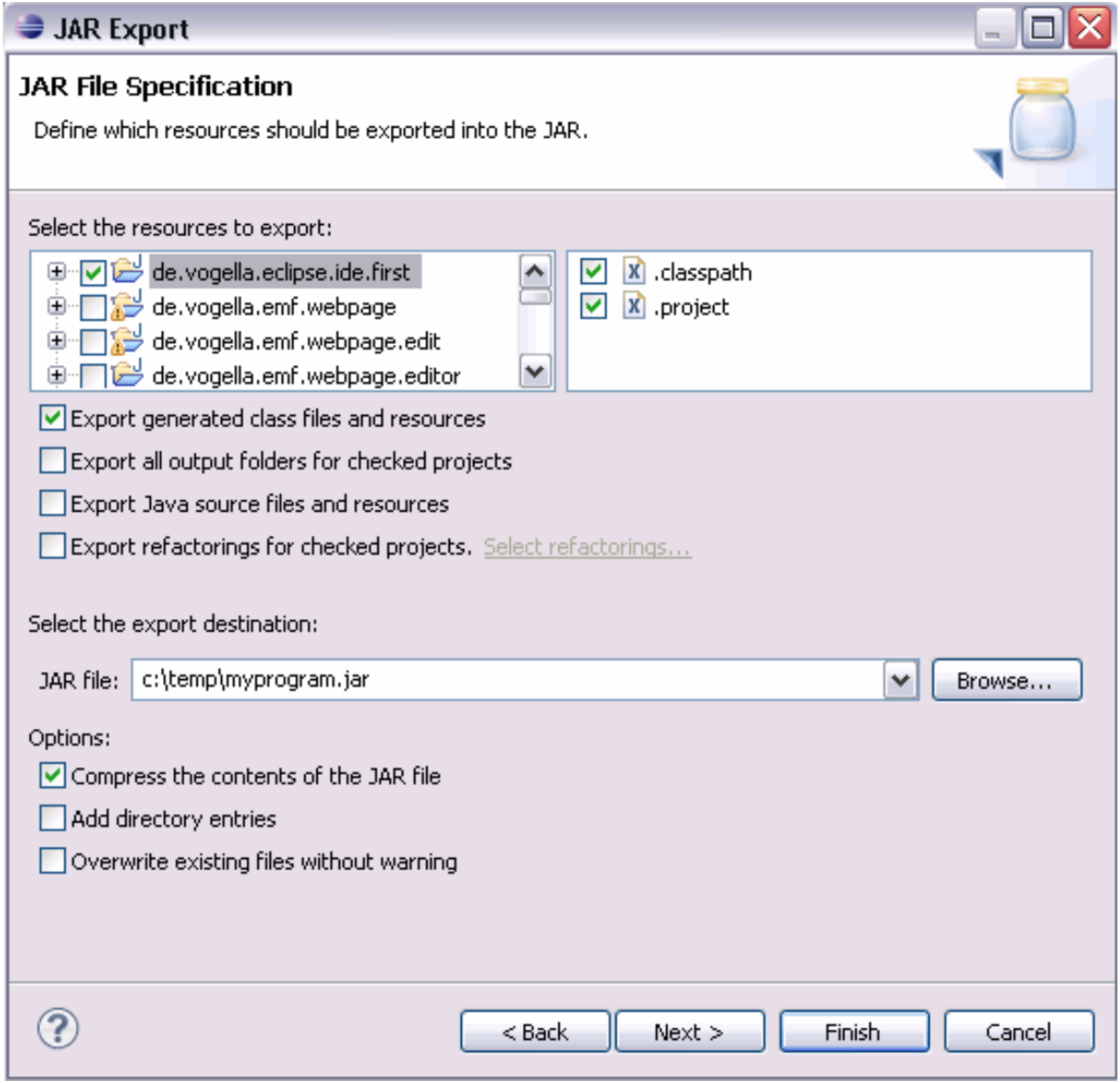


Aby uruchomić program poza środowiskiem Javy należy przygotować plik .jar (jest to domyślne rozszerzenie dla aplikacji Javy). Klikamy na projekt prawym przyciskiem myszy i wybieramy Export.



Wybieramy Java JAR file, next po czym wpisujemy ścieżkę zapisu i nazwę pliku .jar i kończymy za pomocą przycisku Finish.




JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
de.vogella.eclipse.ide.first
de.vogella.emf.webpage
de.vogella.emf.webpage.edit
de.vogella.emf.webpage.editor
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file: c:\temp\myprogram.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel

Asystent treści (content assist), szybkie naprawy i nawigacja po klasach

Asystent treści

Asystent treści wywoływany jest za pomocą kombinacji klawiszy CTRL+SPACE i umożliwia wyświetlenie pomocy w edytorze dostosowując swoją treść do aktualnego kodu. Np. wpisując „syso" i używając asystenta treści ten nam automatycznie zmieni kod na „System.out.println("")".
Tak samo jeżeli posiadamy referencję do obiektu i chcemy zobaczyć jego metody naciskamy CTRL+SPACE.

Szybkie naprawy

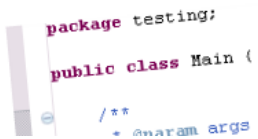
Za każdym razem, gdy Eclipse napotka problem podkreśli problematyczny kawałek kodu w edytorze. Najeżdżając kursorem myszy bądź też znakiem zachęty na podkreślony na czerwono tekst i używając kombinacji klawiszy CTRL+1 wywołujemy okno szybkiej naprawy.
Jest to naprawdę potężne narzędzie, pozwalające np. na szybkie tworzenie brakujących zmiennych lokalnych, metod i klas.

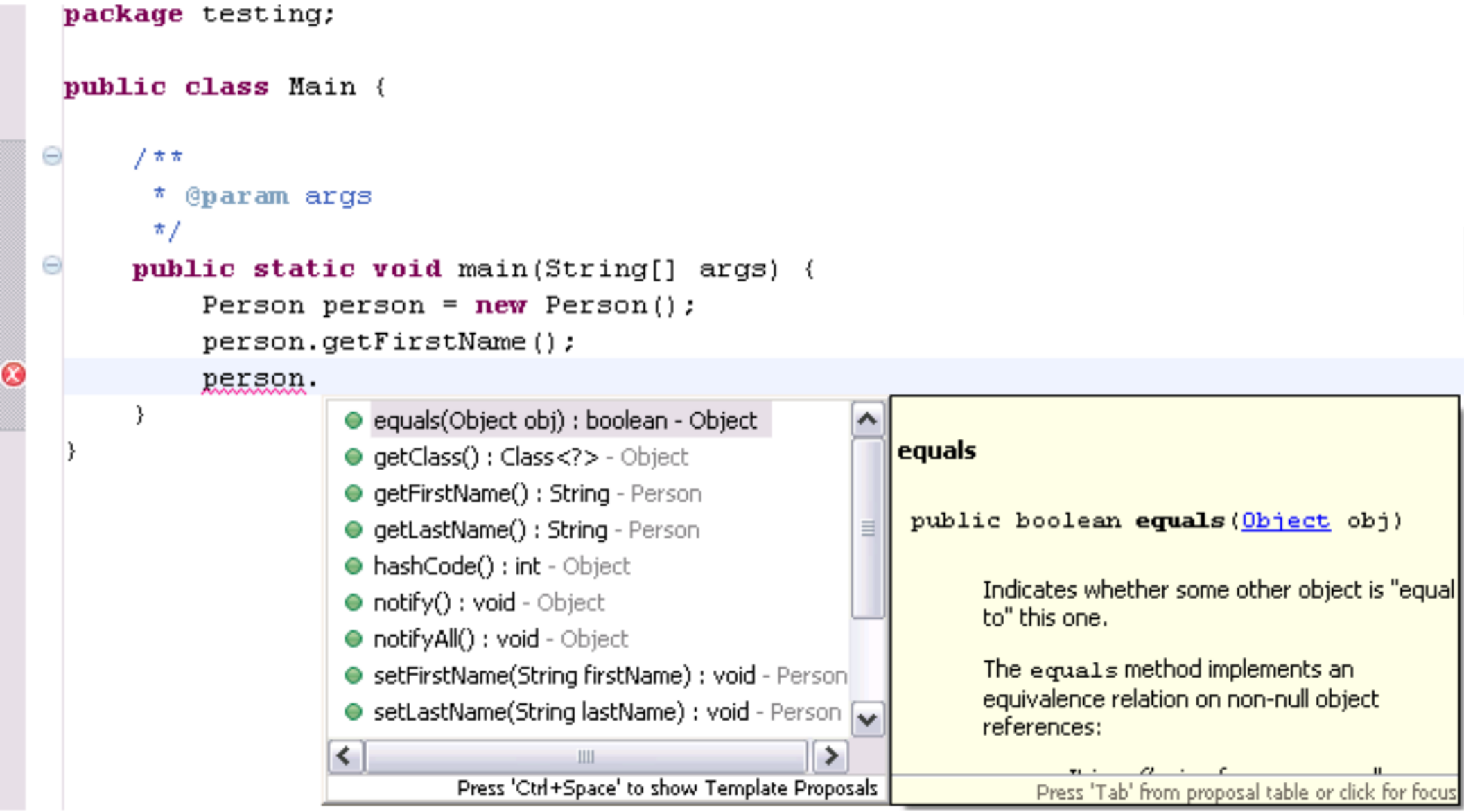
Nawigacja po klasach

Można nawigować pomiędzy klasami używając eksploratora paczek. Istnieją również alternatywne metody. Można przejść do danej klasy po najechaniu na nią w kodzie i użycia przycisku F3. Istnieje też możliwość otworzenia okna do wyszukiwania klas za pomocą kombinacji klawiszy CTRL+SHIFT+T.

Asystent treści

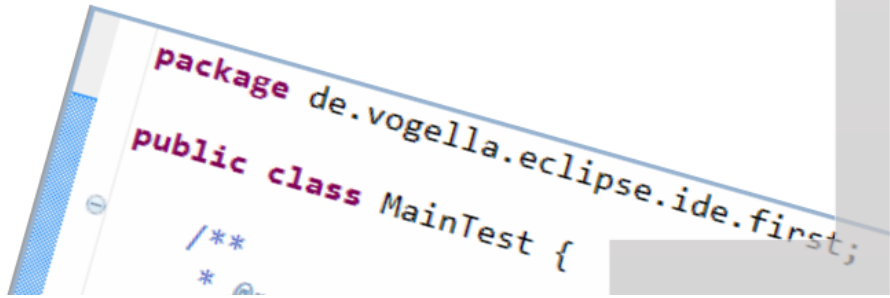
Asystent treści wywoływany jest za pomocą kombinacji klawiszy CTRL+SPACE i umożliwia wyświetlenie pomocy w edytorze dostosowując swoją treść do aktualnego kodu. Np. wpisując „syso” i używając asystenta treści ten nam automatycznie zmieni kod na „System.out.println("")”.
Tak samo jeżeli posiadamy referencję do obiektu i chcemy zobaczyć jego metody naciskamy CTRL+SPACE.

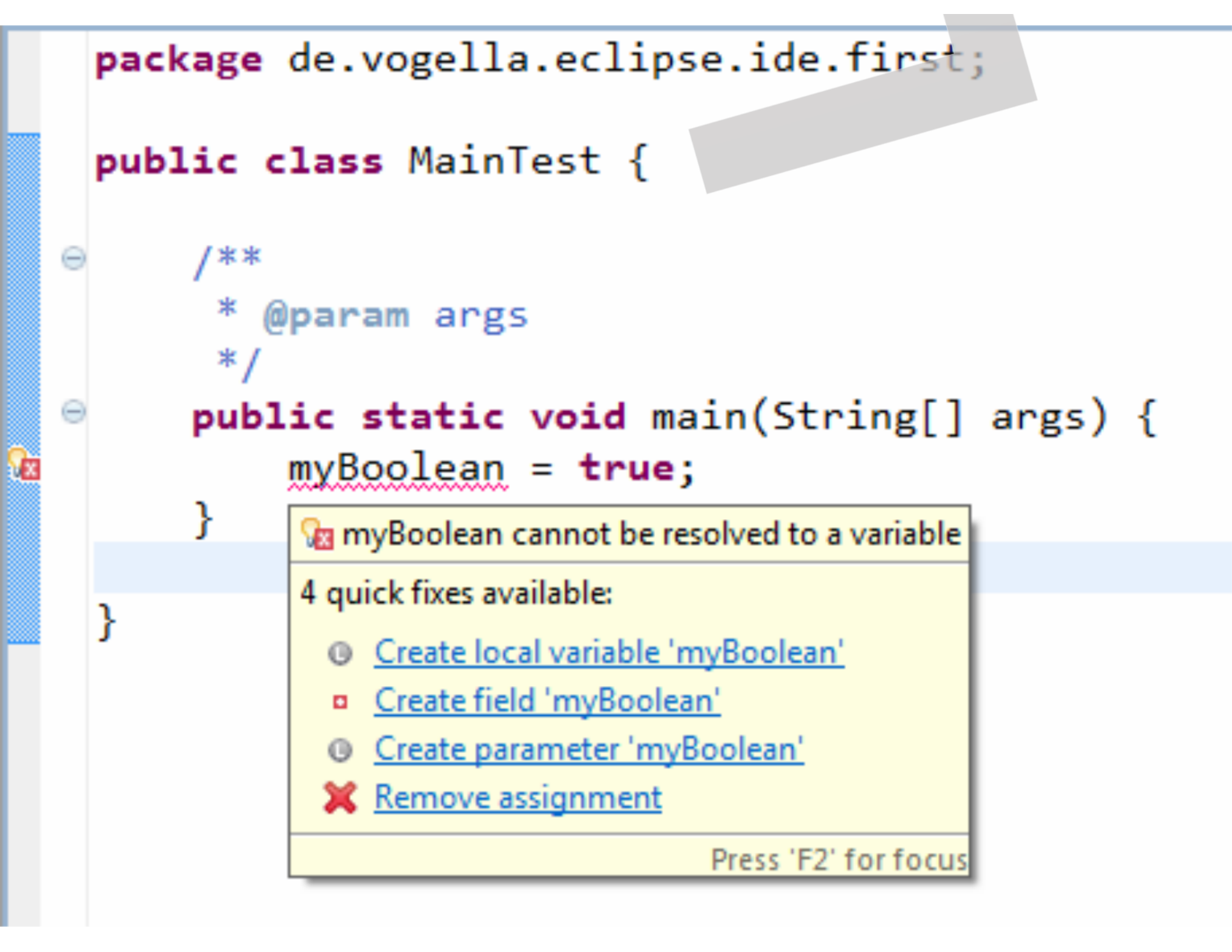



package testing;
public class Main {
/**
* @param args
*/
public static void main(String[] args) {
Person person = new Person();
person.getFirstName();
person.
}
}
equals(Object obj) : boolean - Object
getClass() : Class<?> - Object
getFirstName() : String - Person
getLastName() : String - Person
hashCode() : int - Object
notify() : void - Object
notifyAll() : void - Object
setFirstName(String firstName) : void - Person
setLastName(String lastName) : void - Person
Press 'Ctrl+Space' to show Template Proposals
equals
public boolean equals(Object obj)
Indicates whether some other object is "equal to" this one.
The equals method implements an equivalence relation on non-null object references:
Press 'Tab' from proposal table or click for focus

Szybkie naprawy

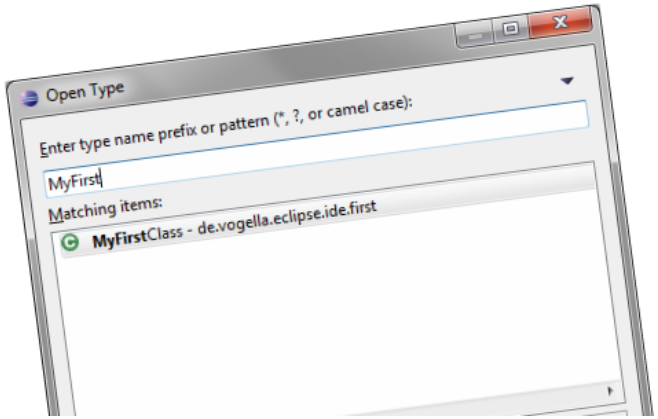
Za każdym razem, gdy Eclipse napotka problem podkreśli problematyczny kawałek kodu w edytorze. Najeżdżając kursorem myszy bądź też znakiem zachęty na podkreślony na czerwono tekst i używając kombinacji klawiszy CTRL+1 wywołujemy okno szybkiej naprawy.
Jest to naprawdę potężne narzędzie, pozwalające np. na szybkie tworzenie brakujących zmiennych lokalnych, metod i klas.

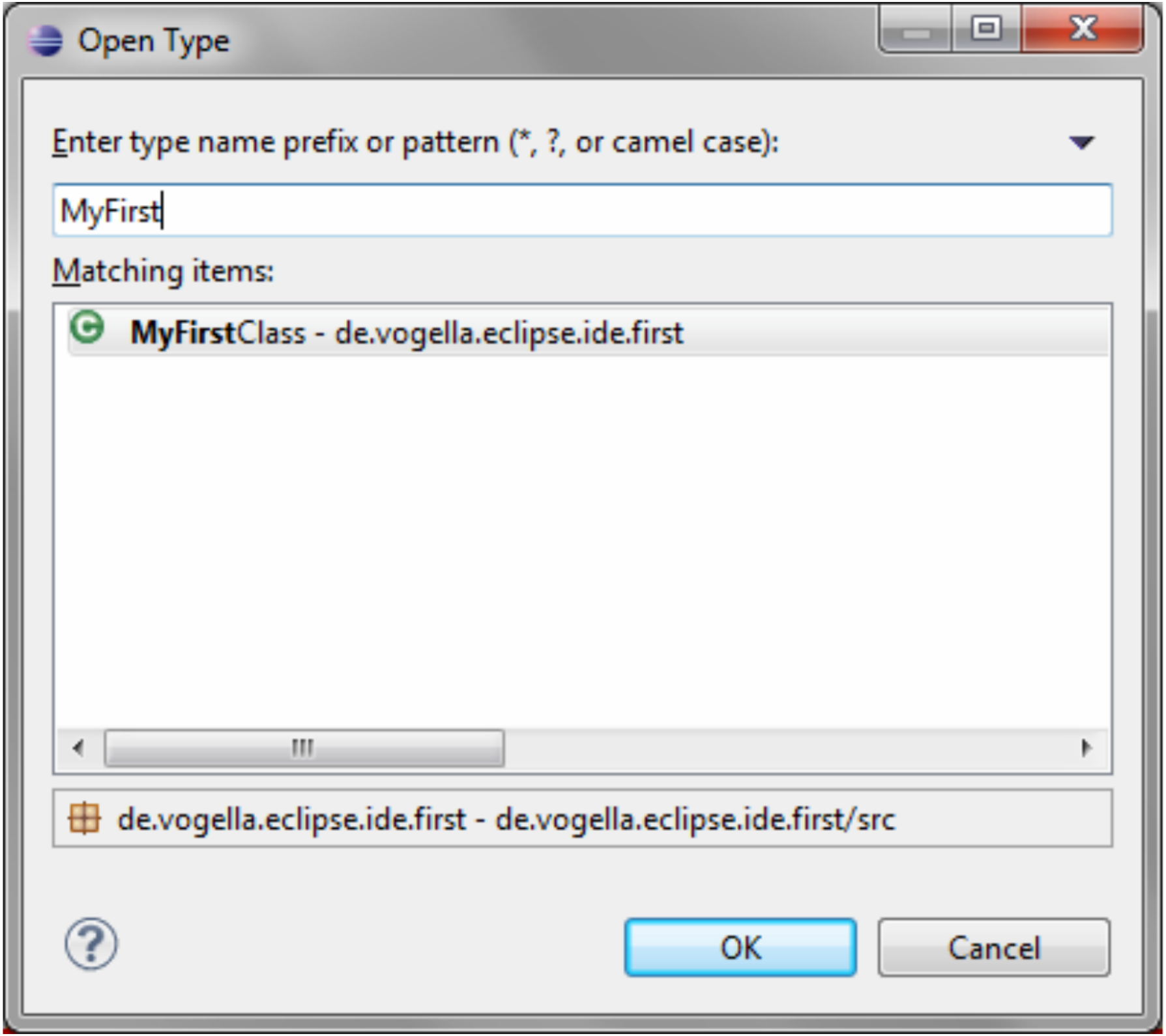



package de.vogella.eclipse.ide.first;
public class MainTest {
/**
* @param args
*/
public static void main(String[] args) {
myBoolean = true;
}
}
myBoolean cannot be resolved to a variable
4 quick fixes available:
Create local variable 'myBoolean'
Create field 'myBoolean'
Create parameter 'myBoolean'
Remove assignment
Press 'F2' for focus

Nawigacja po klasach

Można nawigować pomiędzy klasami używając eksploratora paczek. Istnieją również alternatywne metody. Można przejść do danej klasy po najechaniu na nią w kodzie i użycia przycisku F3. Istnieje też możliwość otworzenia okna do wyszukiwania klas za pomocą kombinacji klawiszy CTRL+SHIFT+T.

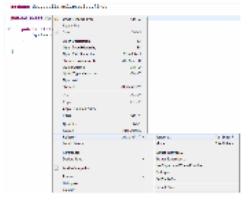



Open Type
Enter type name prefix or pattern (*, ?, or camel case):
MyFirst
Matching items:
MyFirstClass - de.vogella.eclipse.ide.first
de.vogella.eclipse.ide.first - de.vogella.eclipse.ide.first/src
OK
Cancel

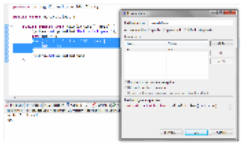
Refaktoryzacja

Refaktoryzacja jest procesem restrukturyzacji kodu bez zmieniania jego zachowania, np. zmienianie nazw klas lub metod.

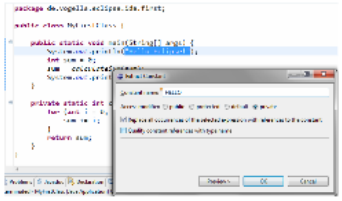
Eclipse wspiera proste metody refaktoryzacji jak np. zmiana nazw czy przenoszenie plików. Można na przykład zaznaczyć klasę, kliknąć na nią prawym przyciskiem myszy i wybrać Refactor Rename. Eclipse również zmieni nazwę wszystkim wywołaniom tej klasy.



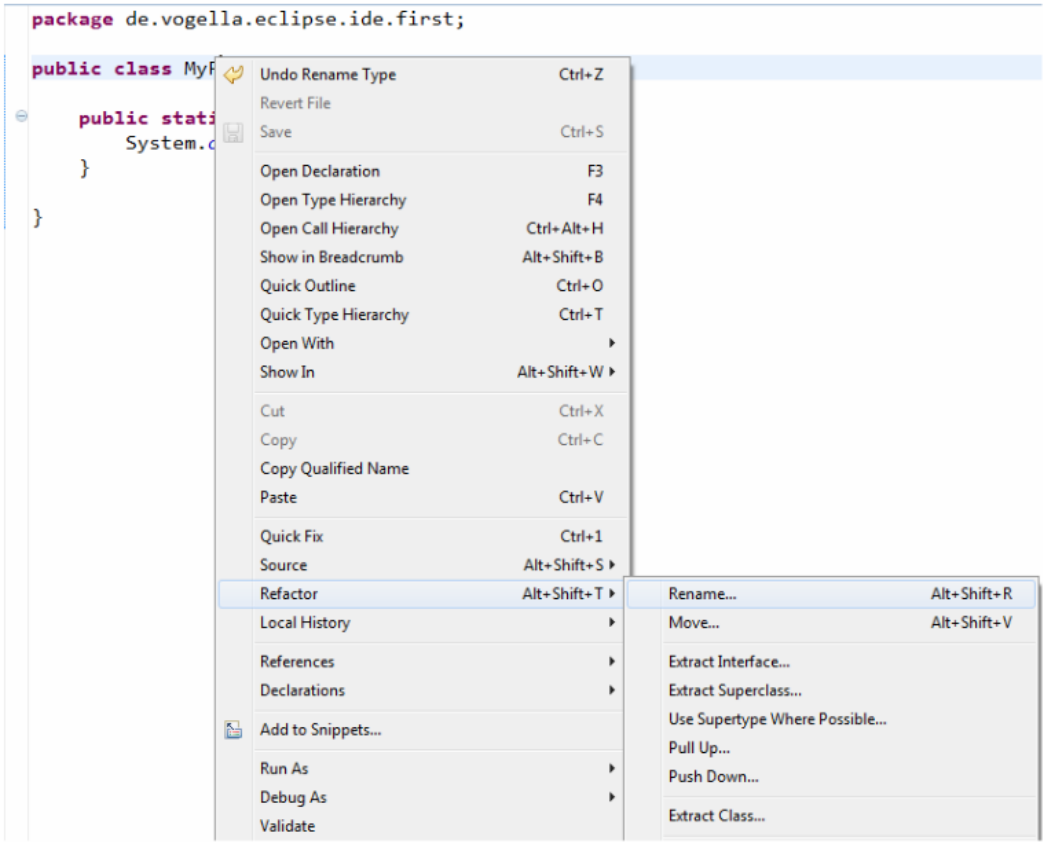
Innym przydatną metodą do refaktoryzacji jest tworzenie metod z zaznaczonego kodu. W tym celu zaznaczamy interesujący nas kawałek kodu, klikamy PPM i wybieramy Refactoring Extract Method gdzie możemy wpisać nazwę naszej metody.



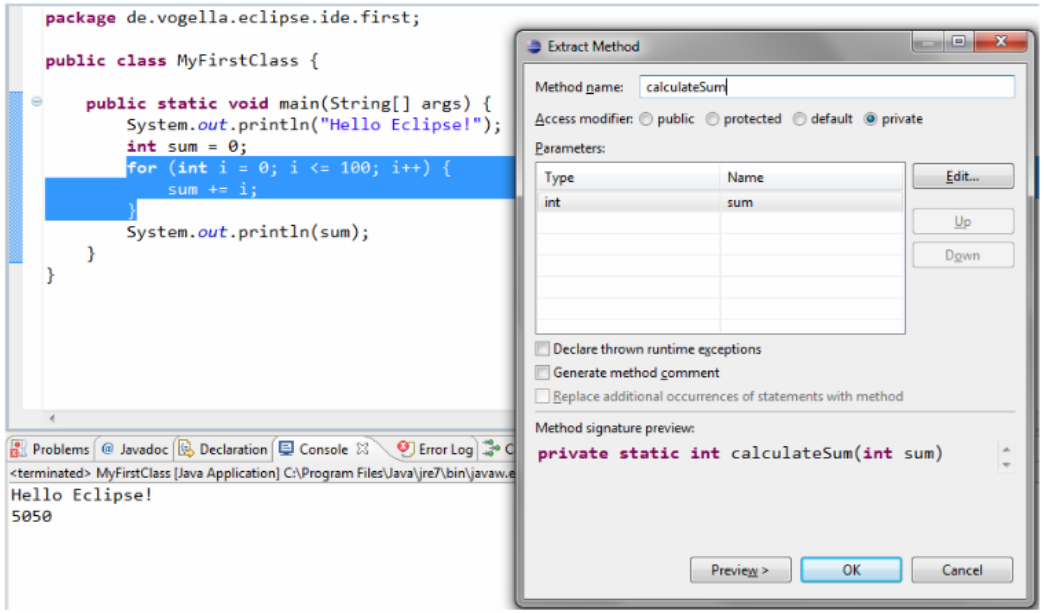
Eclipse umożliwia też tworzenie stałych z wartości. Zaznaczamy wybrany element w kodzie, PPM, Refactor Extract Constant i wpisujemy nazwę stałej wartości.



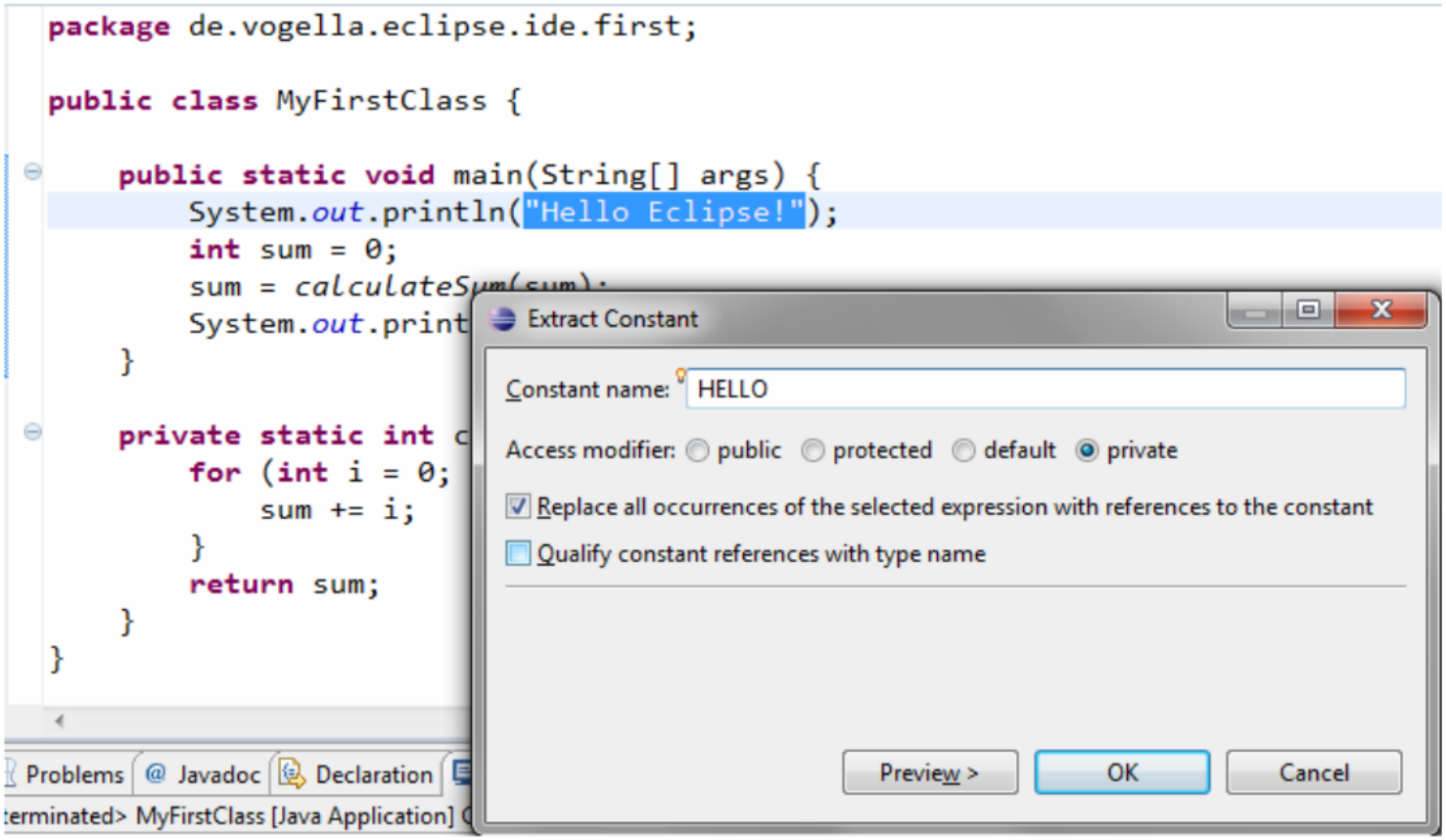
Eclipse wspiera proste metody refaktoryzacji jak np. zmiana nazw czy przenoszenie plików. Można na przykład zaznaczyć klasę, kliknąć na nią prawym przyciskiem myszy i wybrać Refactor Rename. Eclipse również zmieni nazwę wszystkim wywołaniom tej klasy.



Innym przydatną metodą do refaktoryzacji jest tworzenie metod z zaznaczonego kodu. W tym celu zaznaczamy interesujący nas kawałek kodu, klikamy PPM i wybieramy Refactoring Extract Method gdzie możemy wpisać nazwę naszej metody.



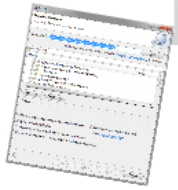
Eclipse umożliwia też tworzenie stałych z wartości. Zaznaczamy wybrany element w kodzie, PPM, Refactor Extract Constant i wpisujemy nazwę stałej wartości.



Aktualizacje i instalacja pluginów

Menadżer aktualizacji Eclipse

Eclipse zawiera menadżer aktualizacji który pozwala na instalowanie i aktualizacje komponentów. Aby zaktualizować Eclipse wybieramy Help Check for Updates. Program poszuka aktualizacji dla zainstalowanych komponentów. Jeżeli je znajdzie to zapyta się o potwierdzenie aktualizacji.
Aby zainstalować nowe elementy należy wybrać Help Install New Software. Tam z listy „Work with" należy wybrać (bądź też wpisać) URL z którego chcemy zainstalować dodatki.



Ręczna instalacja pluginów z użyciem katalogu „dropins"

Może się zdarzyć tak, że dodatki nie posiadają swojej strony współdziałającej z instalatorem Eclipse. Należy wtedy pliki .jar pluginu skopiować do katalogu „dropins" w głównym katalogu środowiska i zrestartować Eclipse.

Eclipse Marketplace

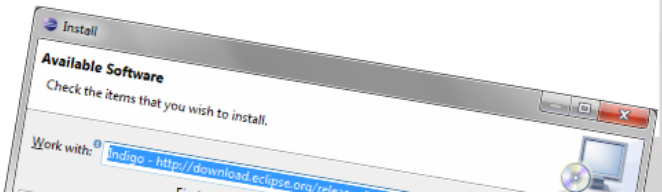
Eclipse zawiera również klienta, który umożliwia instalowanie komponentów prosto z Eclipse Marketplace. Niewątpliwą zaletą tego rozwiązania jest możliwość szukiwania dostępnych pluginów, odkrywania popularnych rozszerzeń i podpatrywanie opisu i oceny.
W porównaniu do menadżera aktualizacji nie trzeba ręcznie wpisywać adresu URL.
Aby otworzyć Eclipse Marketplace wybieramy Help Eclipse Markecplace

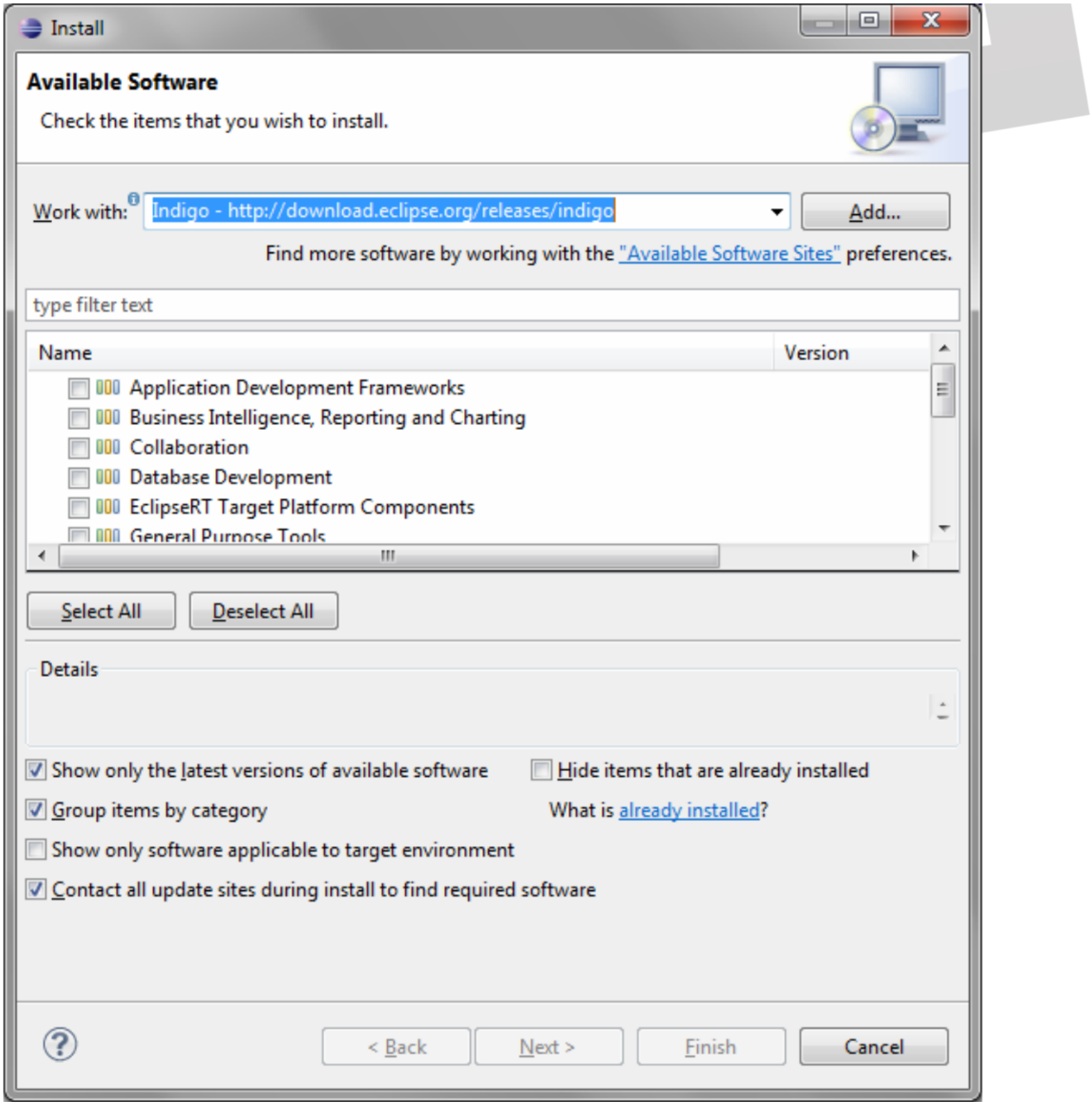
Wymagany restart?

Po aktualizacji bądź też instalacji nowych komponentów dobrym pomysłem jest ponowne uruchomienie Eclipse. Niestety może się zdarzyć, że jakieś dodatki nie będą działać poprawnie.

Menadżer aktualizacji Eclipse

Eclipse zawiera menadżer aktualizacji który pozwala na instalowanie i aktualizacje komponentów. Aby zaktualizować Eclipse wybieramy Help Check for Updates. Program poszuka aktualizacji dla zainstalowanych komponentów. Jeżeli je znajdzie to zapyta się o potwierdzenie aktualizacji.
Aby zainstalować nowe elementy należy wybrać Help Install New Software. Tam z listy „Work with" należy wybrać (bądź też wpisać) URL z którego chcemy zainstalować dodatki.



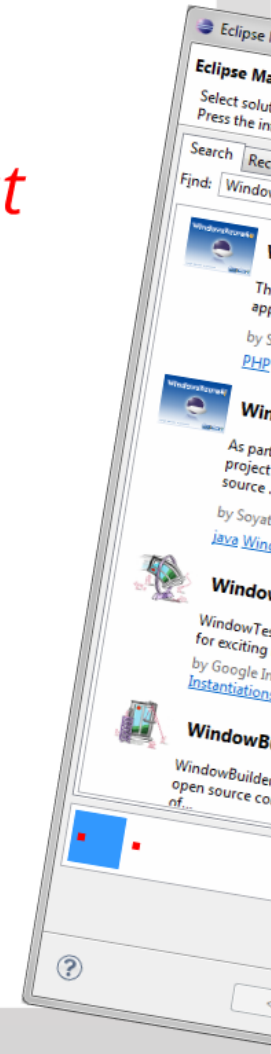

Install
Available Software
Check the items that you wish to install.
Work with:
Indigo - http://download.eclipse.org/releases/indigo
Add...
Find more software by working with the "Available Software Sites" preferences.
type filter text
Name
Version
Application Development Frameworks
Business Intelligence, Reporting and Charting
Collaboration
Database Development
EclipseRT Target Platform Components
General Purpose Tools
Select All
Deselect All
Details
Show only the latest versions of available software
Hide items that are already installed
Group items by category
What is already installed?
Show only software applicable to target environment
Contact all update sites during install to find required software
< Back
Next >
Finish
Cancel

Ręczna instalacja pluginów z użyciem katalogu „dropins”

Może się zdarzyć tak, że dodatki nie posiadają swojej strony współdziałającej z instalatorem Eclipse. Należy wtedy pliki .jar pluginu skopiować do katalogu „dropins” w głównym katalogu środowiska i zrestartować Eclipse.

Eclipse Marketplace

Eclipse zawiera również klienta, który umożliwia instalowanie komponentów prosto z Eclipse Marketplace. Niewątpliwą zaletą tego rozwiązania jest możliwość szukiwania dostępnych pluginów, odkrywania popularnych rozszerzeń i podpatrywanie opisu i oceny.
W porównaniu do menadżera aktualizacji nie trzeba ręcznie wpisywać adresu URL.
Aby otworzyć Eclipse Marketplace wybieramy Help Eclipse Markecplace

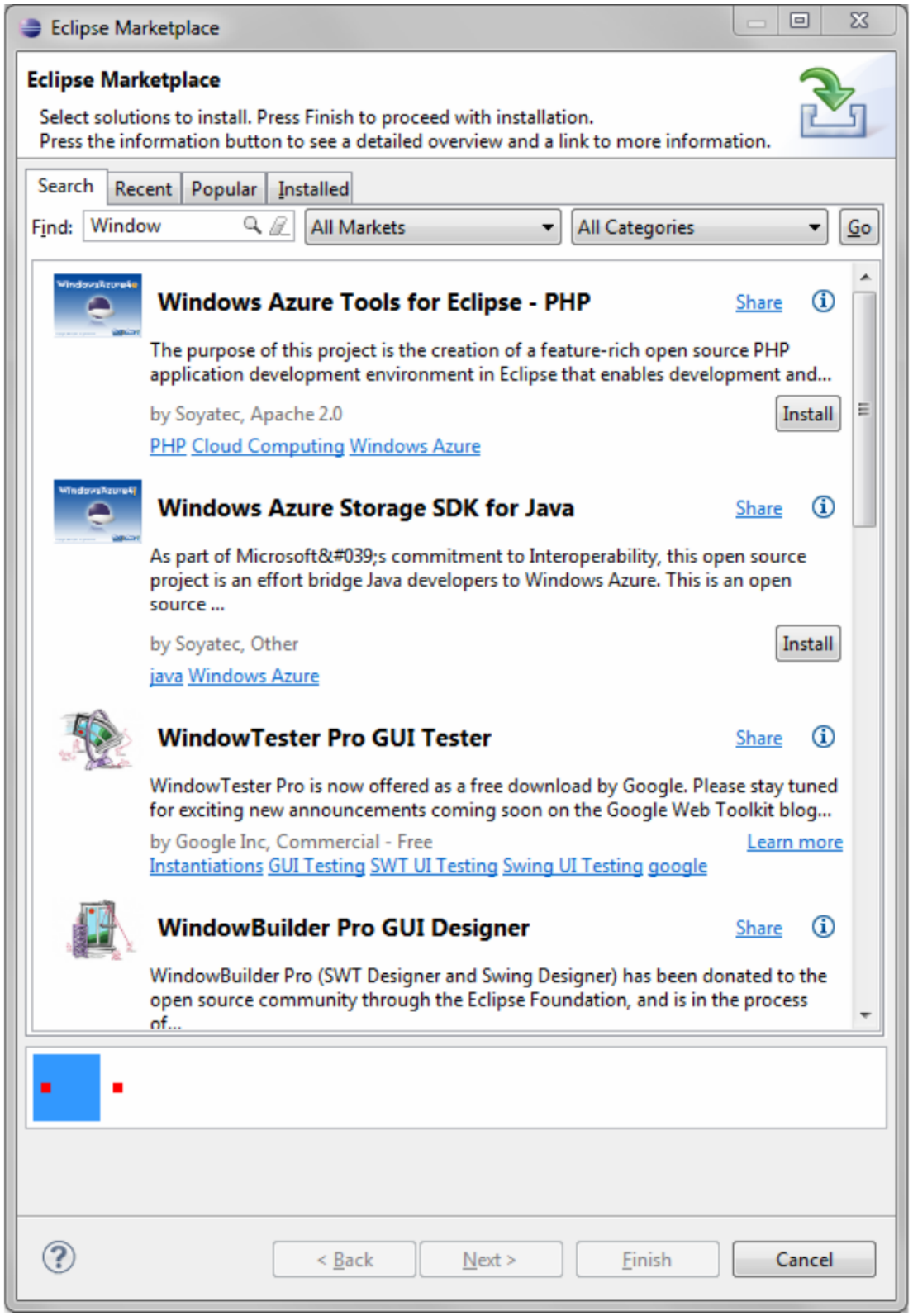


nia jest

ywanie

trzeba

y Help



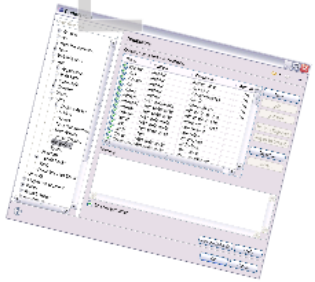
Wymagany restart?

Po aktualizacji bądź też instalacji nowych komponentów dobrym pomysłem jest ponowne uruchomienie Eclipse. Niestety może się zdarzyć, że jakieś dodatki nie będą działać poprawnie.

Ustawienia zwiększające efektywność

Szablony

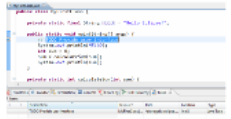
Jeżeli często pisze się dany kawałek kodu warto stworzyć szablon, który będzie aktywowany przez autouzupełnienie (CTRL+SPACE).
Na przykład, jeżeli chcemy stworzyć szablon do metody public void name(){} to w edytorze szablonów (Window Preferences Java Editor Templates) wybieramy New i odpowiednio uzupełniamy. Teraz pisząc tylko npm i używając autouzupełnienia stworzy nam się cały kawałek kodu.



Zarządzanie zadaniami

Można używać komentarzy //TODO w kodzie aby dodawać przypomnienie do zadań. Zadania te można znaleźć w widoku Tasks w Eclipse (Window Show View Tasks). Po podwójnym kliknięciu na zadanie programista jest przenoszony do konkretnego kawałka kodu.

// TODO Zrób coś

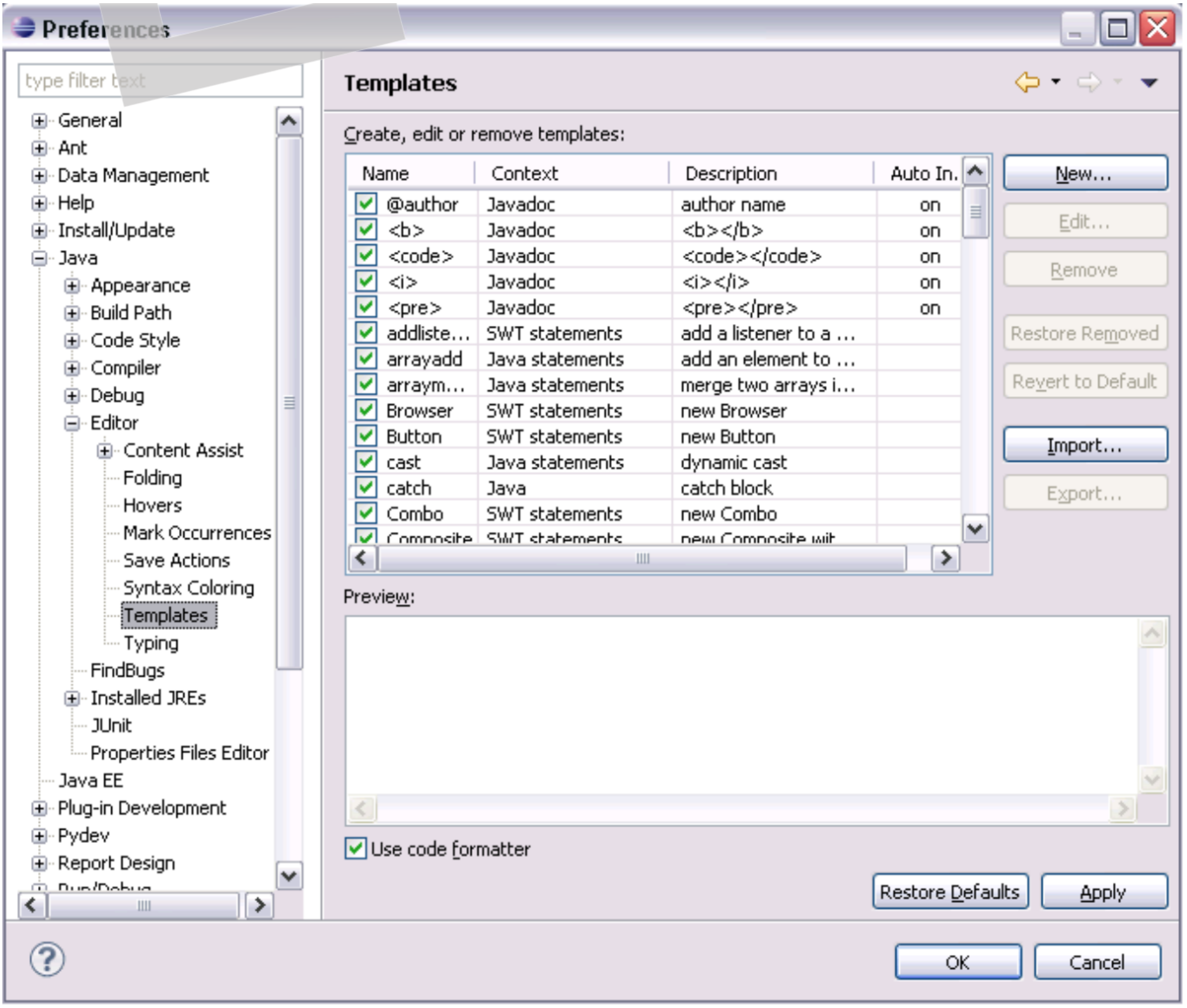


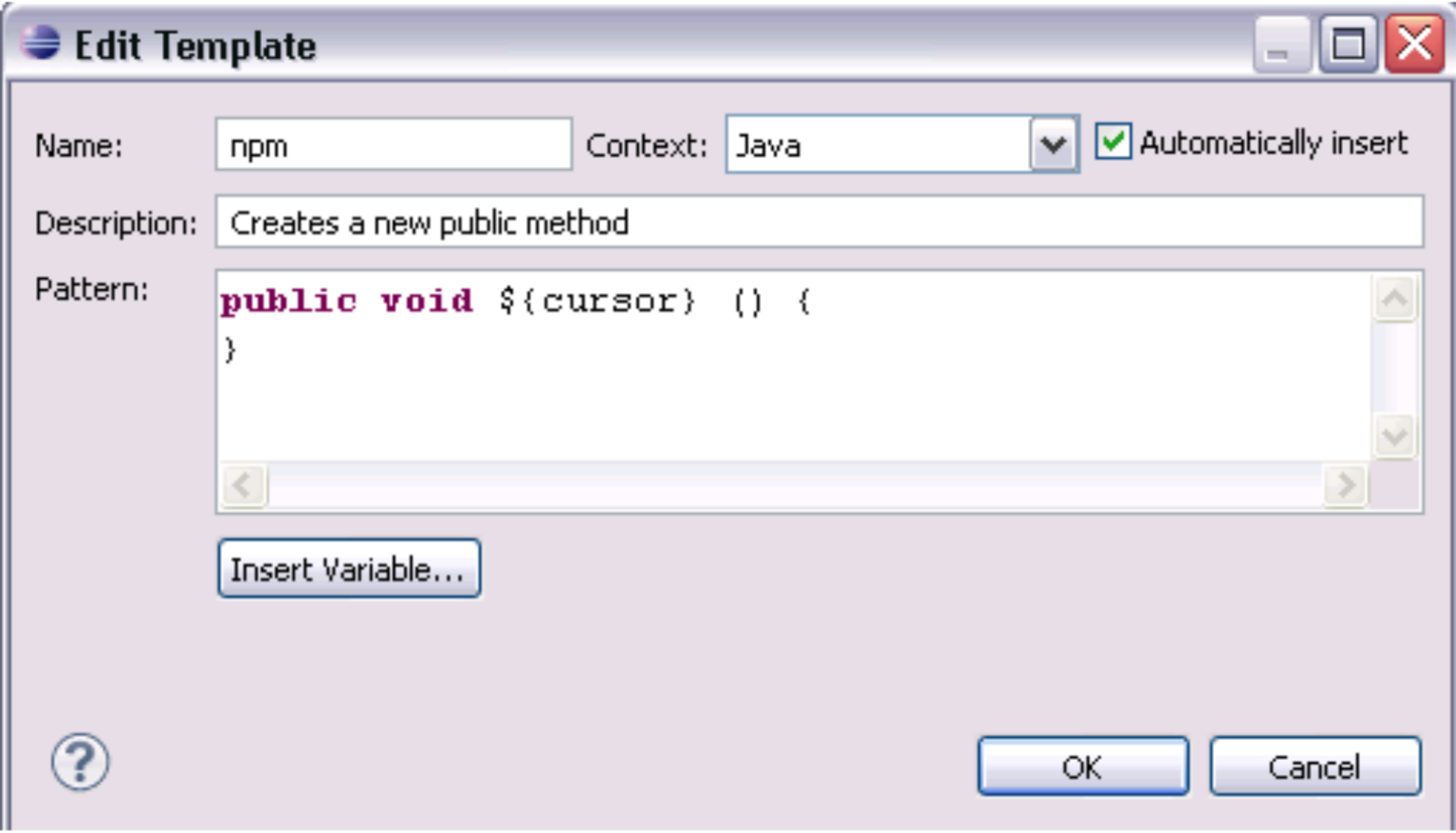
Zestawy robocze

Z czasem w eksploratorze paczek znajduje się coraz więcej projektów. Jest więc coraz trudniej znaleźć potrzebne informacje.

Szablony

Jeżeli często pisze się dany kawałek kodu warto stworzyć szablon, który będzie aktywowany przez autouzupełnienie (CTRL+SPACE).
Na przykład, jeżeli chcemy stworzyć szablon do metody public void name(){} to w edytorze szablonów (Window Preferences Java Editor Templates) wybieramy New i odpowiednio uzupełniamy. Teraz pisząc tylko npm i używając autouzupełnienia stworzy nam się cały kawałek kodu.

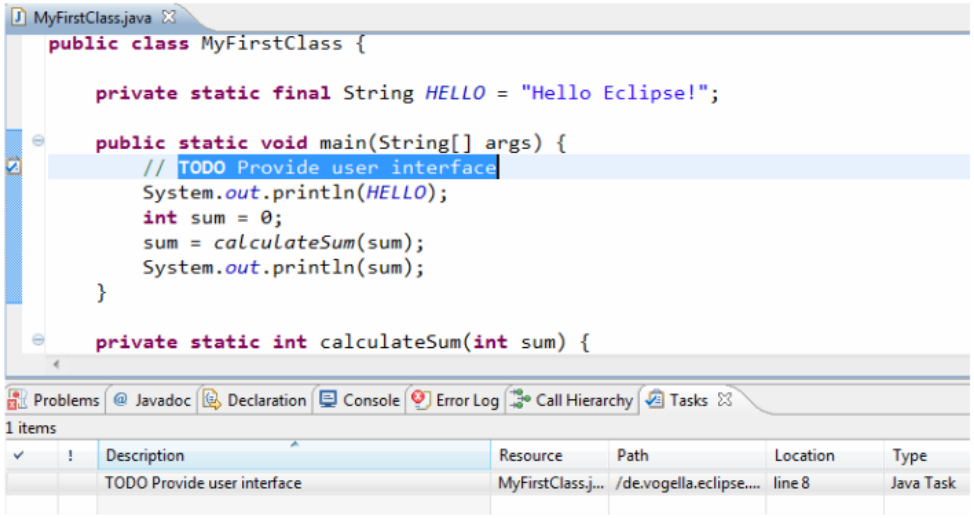

Preferences
type filter text
General
Ant
Data Management
Help
Install/Update
Java
Appearance
Build Path
Code Style
Compiler
Debug
Editor
Content Assist
Folding
Hovers
Mark Occurrences
Save Actions
Syntax Coloring
Templates
Typing
FindBugs
Installed JREs
JUnit
Properties Files Editor
Java EE
Plug-in Development
Pydev
Report Design
Templates
Create, edit or remove templates:
Name
Context
Description
Auto In.
@author Javadoc author name on
 Javadoc on
<code> Javadoc <code></code> on
 Javadoc on
<pre> Javadoc <pre></pre> on
addliste... SWT statements add a listener to a ...
arrayadd Java statements add an element to ...
arraym... Java statements merge two arrays i...
Browser SWT statements new Browser
Button SWT statements new Button
cast Java statements dynamic cast
catch Java catch block
Combo SWT statements new Combo
New...
Edit...
Remove
Restore Removed
Revert to Default
Import...
Export...
Preview:
Use code formatter
Restore Defaults
Apply
OK
Cancel


Edit Template
Name:
npm
Context:
Java
Automatically insert
Description:
Creates a new public method
Pattern:
public void ${cursor} () {
}
Insert Variable...
OK
Cancel

Zarządzanie zadaniami

Można używać komentarzy //TODO w kodzie aby dodawać przypomnienie do zadań. Zadania te można znaleźć w widoku Tasks w Eclipse (Window Show View Tasks). Po podwójnym kliknięciu na zadanie programista jest przenoszony do konkretnego kawałka kodu.

// TODO Zrób coś



ny do konkretnego kawałka

MyFirstClass.java

```
public class MyFirstClass {

    private static final String HELLO = "Hello Eclipse!";

    public static void main(String[] args) {
        // TODO Provide user interface
        System.out.println(HELLO);
        int sum = 0;
        sum = calculateSum(sum);
        System.out.println(sum);
    }

    private static int calculateSum(int sum) {
```

Problems | Javadoc | Declaration | Console | Error Log | Call Hierarchy | Tasks

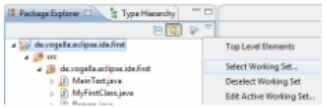
1 items

✓	!	Description	Resource	Path	Location	Type
		TODO Provide user interface	MyFirstClass.j...	/de.vogella.eclipse....	line 8	Java Task

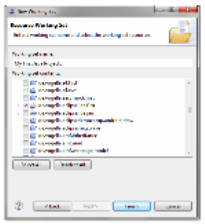
Zestawy robocze

Z czasem w eksploratorze paczek znajduje się coraz więcej projektów. Jest więc coraz trudniej znaleźć potrzebne informacje.

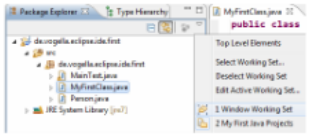
Można temu zapobiec organizując wyświetlane projekty. Aby utworzyć zestaw roboczy należy wybrać Package Explorer ikonka z trójkątem w dół Select Working Set...



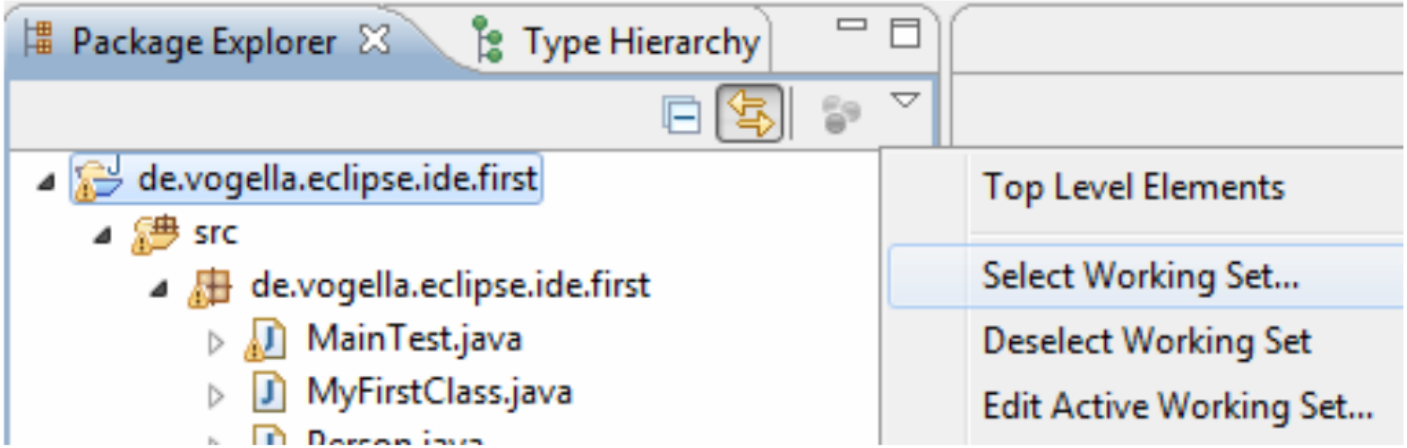
Po wybraniu opcji dodania nowego zestawu roboczego zostaniemy poproszeni o nazwę oraz o wybranie źródeł, które będą się znajdować w tym zestawie.



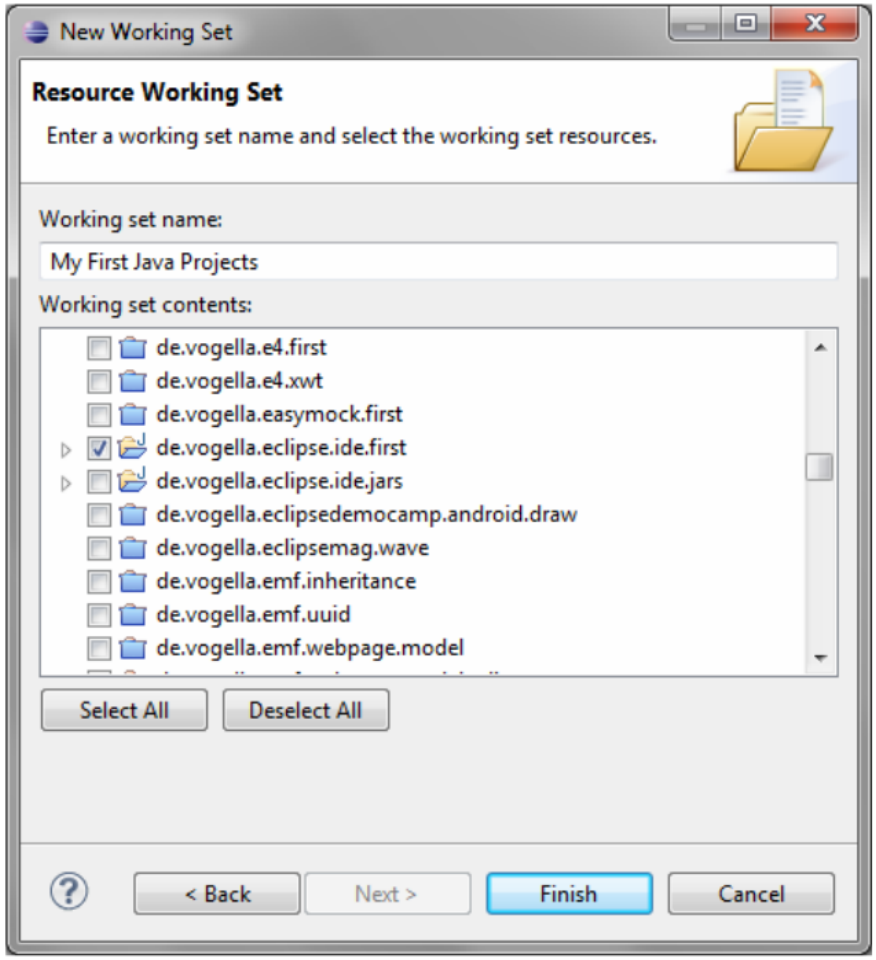
Po zatwierdzeniu operacji można teraz w łatwy sposób z rozwijalnego menu wybierać odpowiedni zestaw roboczy.



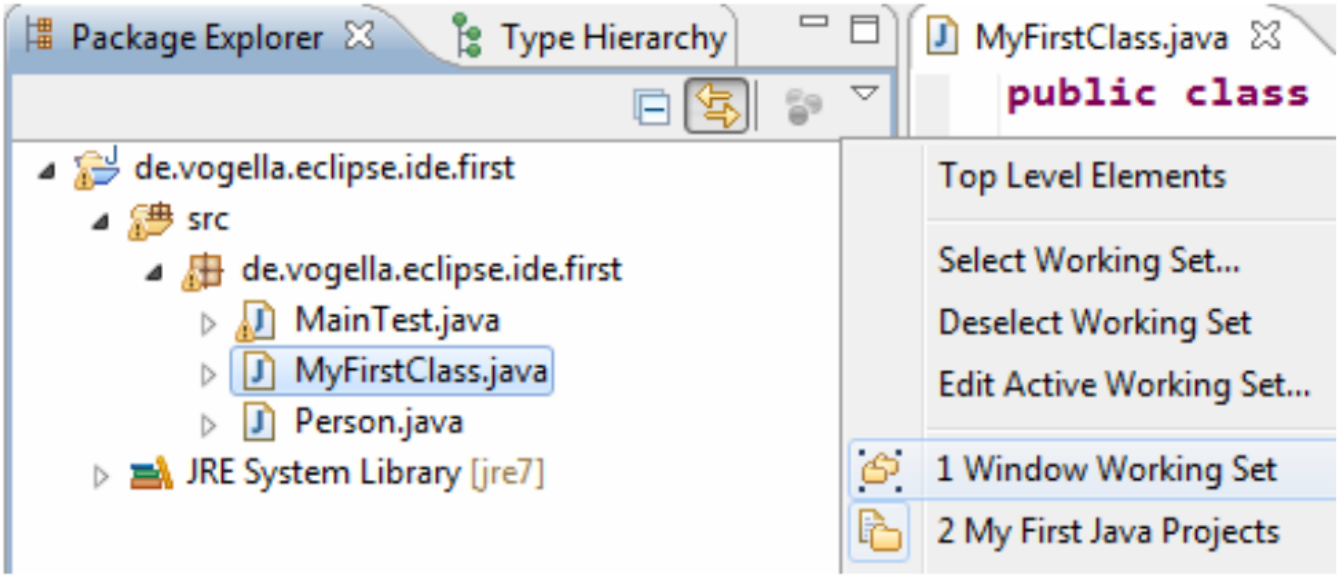
Można temu zapobiec organizując wyświetlane projekty. Aby utworzyć zestaw roboczy należy wybrać Package Explorer ikonka z trójkątem w dół Select Working Set...

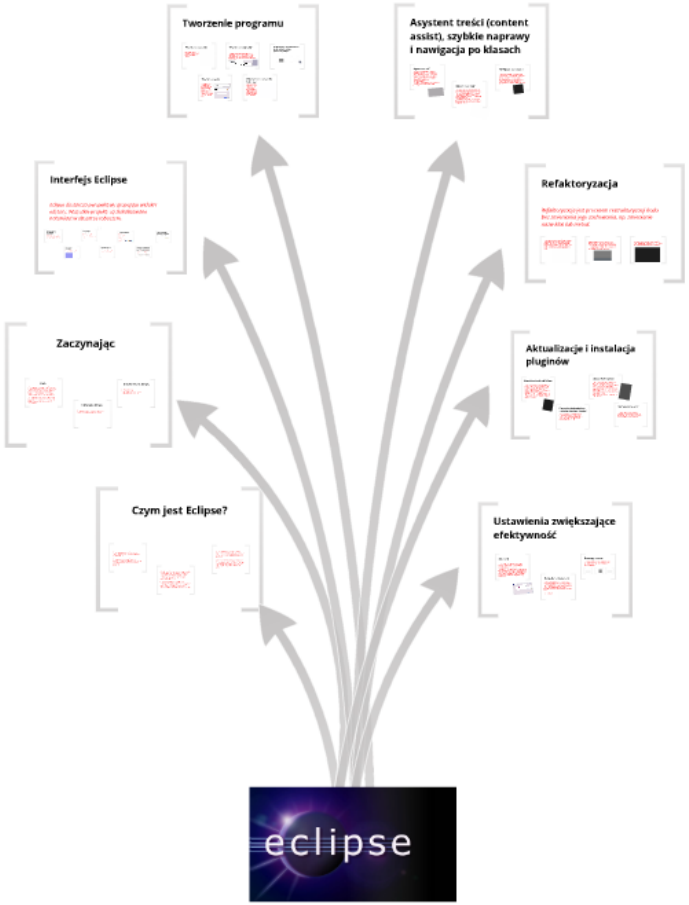


Po wybraniu opcji dodania nowego zestawu roboczego zostaniemy poproszeni o nazwę oraz o wybranie źródeł, które będą się znajdować w tym zestawie.



Po zatwierdzeniu operacji można teraz w łatwy sposób z rozwijalnego menu wybierać odpowiedni zestaw roboczy.





Dziękuję za uwagę :)