

# **Projet de développement**

## **Introduction à Eclipse**

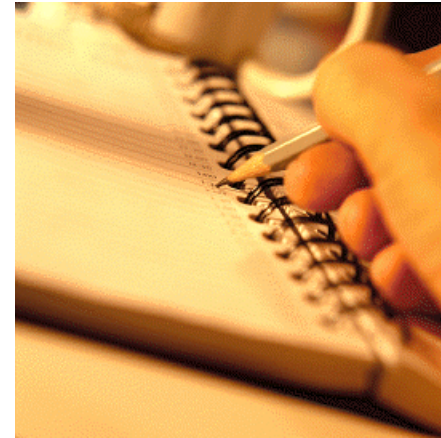
**Philippe Collet**

**Licence 3 MIAGE – S6**

**2013-2014**

# Plan

- ❑ Eclipse : Introduction et historique
- ❑ Architecture d'Eclipse
- ❑ Environnement de développement
- ❑ JDT
- ❑ Support SVN
- ❑ Application à votre projet



# Introduction : objectif

## □ Un Environnement de Développement Intégré (EDI)

- Un logiciel regroupant un ensemble d'outils nécessaires au développement des applications dans un langage de programmation

## □ Objectifs généraux : fournir des fonctionnalités

- un éditeur de texte spécialisé
- un compilateur
- un débogueur
- des outils automatiques de gestion d'applications ayant plusieurs fichiers source (projets)
- un gestionnaire de versions
- un générateur de documentation

# Eclipse et ses concurrents

## ❑ Logiciels libres :

- Emacs, XEmacs : basique, mais adaptables à tout langage
- Kdevelop (KDE) : C, C++, basé sur les outils GNU
- Netbeans (Sun/Oracle) : initialement conçu pour Java, maintenant C, C++, XML et HTML
  - ◆ Au départ plus lourd et lent, maintenant plus léger (et vraisemblablement plus rapide)
- Eclipse (OTI-IBM) : Java, C/C++, PHP, HTML, etc.

## ❑ Logiciels propriétaires :

- IntelliJ / IDEA : excellent concurrent, en constante amélioration
- Visual Studio (Microsoft) : C/C++, .NET, C#, etc.

## ❑ Logiciel libre

- Destiné à l'origine pour le développement en Java
- Conçu sur la base d'un EDI Java (VA4J), Eclipse devient un EDI pour développer des EDIs et d'autres outils

## ❑ Objectif :

- Offrir une plateforme ouverte pour le développement d'applications
- Non dédiée à un langage ou système d'exploitation ou interface graphique
- Facile à comprendre mais aussi facile à étendre
- Paramétrable selon les besoins/goûts du programmeur
- Capable d'automatiser les tâches lourdes du développement
- Ayant une base stable
- Utilisable pour son propre développement (bootstrap-able)
- Promouvant l'utilisation de Java

# Historique d'Eclipse

- ❑ 1996 : IBM rachète OTI, qui développe la suite d'EDI Visual Age (en SmallTalk), et en particulier VA4J
- ❑ 2001 : après un investissement de 40 M\$, IBM lance Eclipse 1
  - Grand succès populaire car suite ouverte et gratuite (licence CPL).
  - Création du consortium Eclipse (IBM, Borland, RedHat, SuSE, Intel,...)
- ❑ 2002 : Eclipse 2.0
- ❑ 2004 : Eclipse 3.0
- ❑ 2006 : Eclipse 3.2 Europa
  - Première release nommée pour stabiliser les références aux sous-projets
- ❑ 2011 : Eclipse 3.7 Indigo
- ❑ 2013 : Eclipse 4.3 Kepler

# Sous-projets Eclipse

- ❑ **Eclipse : architecture et structure de la plateforme**
- ❑ **Eclipse Tools : outils pour permettre l'enrichissement de la plateforme**
  - PDT, CDT sont basés sur ce sous-projet
- ❑ **Eclipse Technology : recherche sur l'évolution de la plateforme**
  - Très actif pour le passage de Eclipse 2.x à Eclipse 3.x
- ❑ **Test and Performance Tools Platform (TPTP) : outils de test et d'analyse**
- ❑ **Business Intelligence and Reporting Tools (BIRT) : outils de génération d'états**
  - Composé de 4 autres sous-projets

- ❑ **Eclipse Modeling : Plusieurs sous-projets dont**
  - EMF (Eclipse Modeling Framework) : pour la manipulation de modèles et projection vers du code
  - UML2 : métamodèle complet d'UML2 pour création d'outils conforme
  
- ❑ **Data Tools Platform (DTP) : Manipulation de source de données (BD relationnelles essentiellement)**
  
- ❑ **Device Software Development Platform : Outils pour plugins de développement dédié aux applications mobiles**
  
- ❑ **Eclipse SOA Tools Platform (STP) : Outil pour le développement d'applications selon des architectures orientées services (web services, standard SCA...)**



## ❑ **Simplissime :**

- Téléchargez l'archive (dédiée au système ou générique)
- Décompactez la dans un répertoire système
- Créez un lien/raccourci vers l'exécutable eclipse...
- Ca roule

## ❑ **Un peu moins simple : quels plugins et comment les installer ?**

- Décompactez le plugin dans le répertoire dédié
- Utilisez la fonction d'update (tutoriaux en ligne)

## ❑ **Moins simple : quelle version prendre ?**

- Classic
- J2EE (Java Entreprise...)
- Etc.

# Constituants

❑ **Eclipse = plateforme + plug-ins**

❑ **Plateforme**

- Un support d'exécution (runtime) indépendant du système d'exploitation (JVM)
- Un ensemble basique de plug-ins extensibles
- De mécanismes (API), règles et outils pour construire de plug-in
- Un moteur pour découvrir, charger et exécuter des plug-ins

❑ **Plug-in = la plus petite unité qui peut être développée et utilisée séparément**

- se connecte à un point précis de la plateforme
- remplit une tâche (pas forcément exécutable)
- offre des points d'extension
- coexiste avec d'autres plug-ins

❑ **instance (feature) = ensemble de plug-ins qui coopèrent pour offrir un EDI**

# Eclipse : support d'exécution

## ❑ Support d'exécution = Platform Runtime

- Exécute la JVM (Java Virtual Machine)
- Définit les points d'extension et le modèle plug-in

## ❑ Modèle de plugin

- point d'extension = interface
- plug-in = interfaces implémentées + archive Jar + interfaces utilisées
- déclaration de plug-in = manifeste (dépendances à l'exécution) + interface (type)

## ❑ Le support d'exécution

- Découvre dynamiquement les plug-ins et maintient une base relative à leur déclaration
- Charge les plug-ins à la demande.
- Met à jour automatiquement des instances (features)

# Environnement de développement

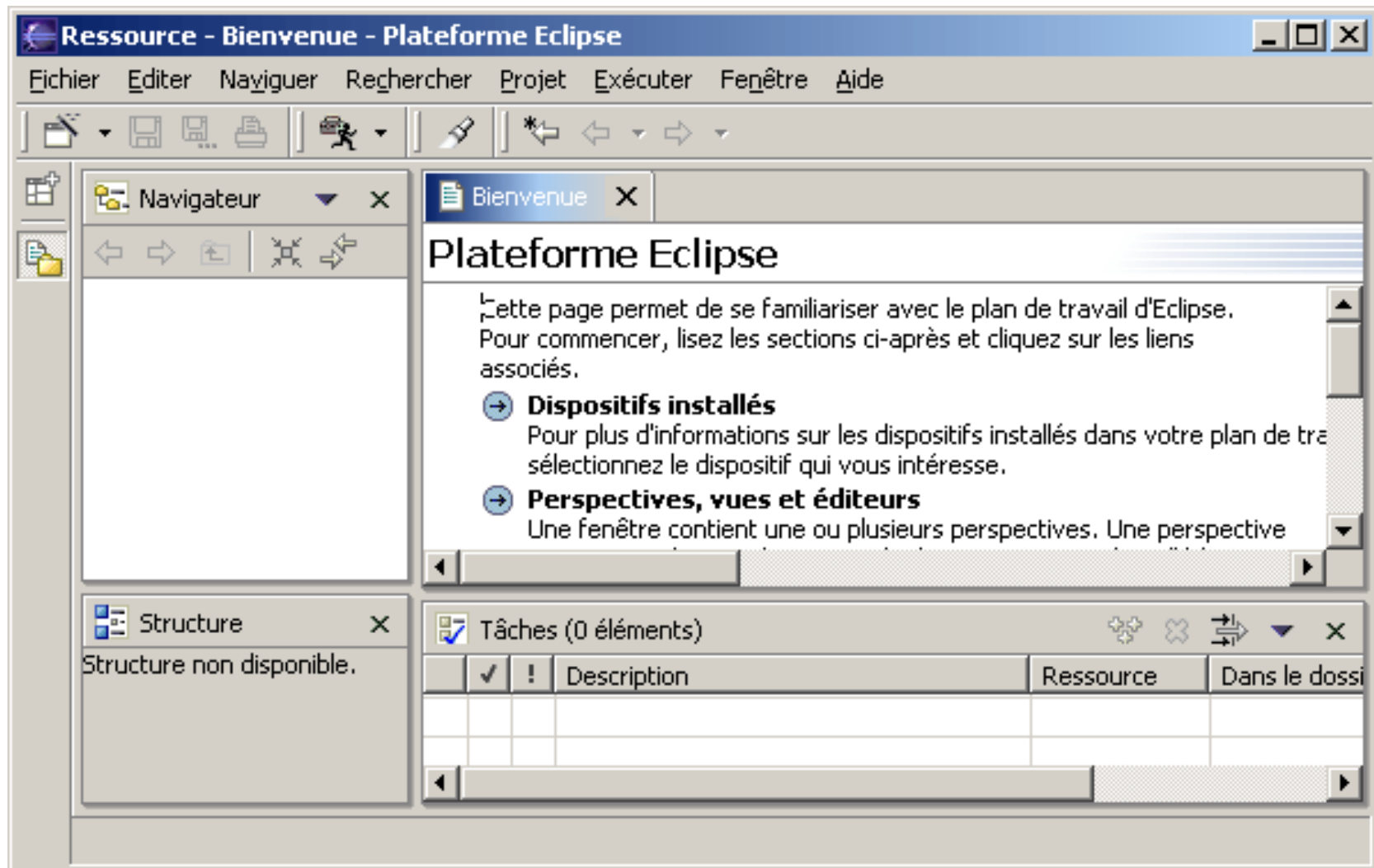
## ❑ Plan de travail = Workbench

- Fournit l'interface visuelle pour l'utilisateur de la plateforme
- (Spécificité Eclipse) : l'interface graphique (UI) a l'apparence d'une application native du système d'exploitation
- est basé sur deux outils (SWT – Standard Widget Tool, JFace) qui peuvent être utilisés directement pour développer des applications

## ❑ Composantes physiques de l'UI : menus, barre d'actions, boutons, onglets, fenêtres

## ❑ Composantes logiques de l'UI (paramétrable par des plug-ins)

# Workbench



# Vues et perspectives

## □ Vue

- fournit des informations sur les objets (structure, composantes, etc.) en communiquant avec d'autres vues ou éditeurs
- Exemple: Navigateur, packages

## □ Editeur

- Edition plus ou moins dédié (langage avec coloration syntaxique, complétion)
- Exemple : éditeur de texte, éditeur Java

## □ Perspective

- ensemble d'éditeurs et vues ayant une disposition précise dans le plan de travail
- Afin de faciliter la réalisation de certaines tâches
- Exemple : navigation, édition Java, exploration de référentiel cvs/svn, synchronisation avec un référentiel
- Le plus étendu des points d'extension

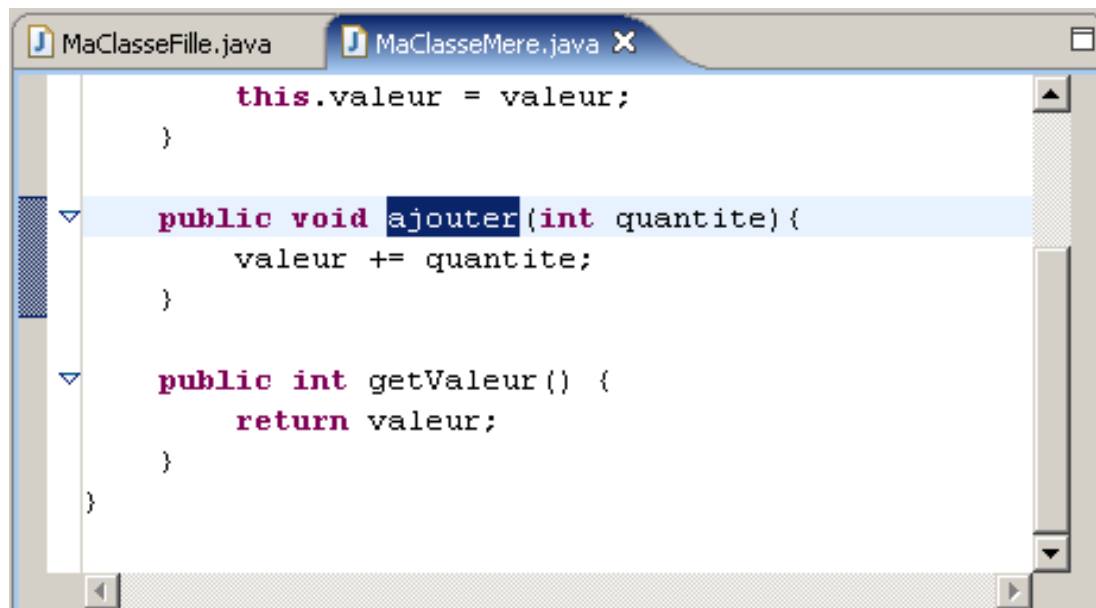
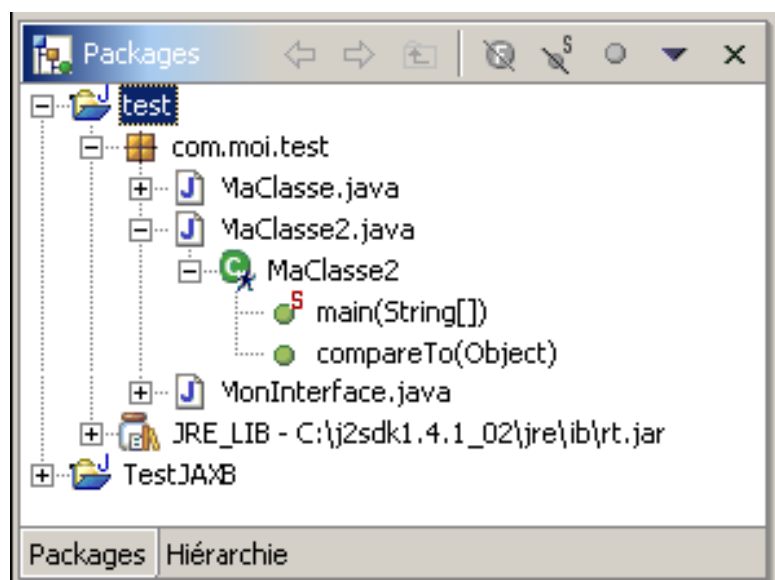
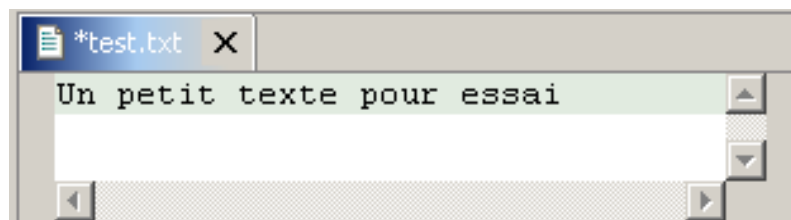
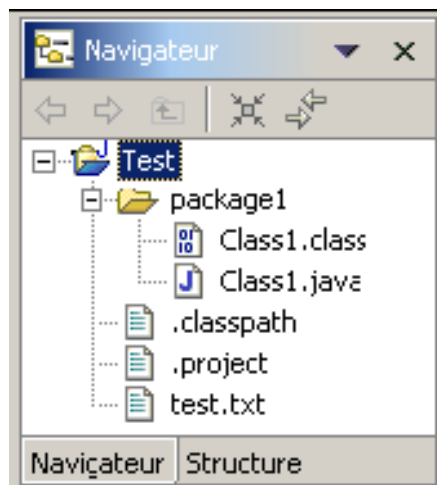
# Perspective

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer (Left):** Shows a project structure with a package `org.eclipse.banking.tests` containing `BankAccountTests.java`.
- Outline (Bottom Left):** Lists the methods `testDeposit()`, `testWithdraw()`, and `testOverdraft()`.
- Editor (Center):** Shows the source code of `BankAccountTests.java`. The code includes imports for `java.math.BigDecimal` and `org.junit.Test`, and defines three test methods: `testDeposit()`, `testWithdraw()`, and `testOverdraft()`. The `testOverdraft()` method is currently selected.
- Problems (Bottom):** Shows a JUnit test run summary: "Finished after 0.031 seconds", "Runs: 3/3", "Errors: 0", and "Failures: 1". A failure trace is visible for `testOverdraft()`, indicating an `AssertionFailedError` because an `InsufficientFundsException` was not thrown.



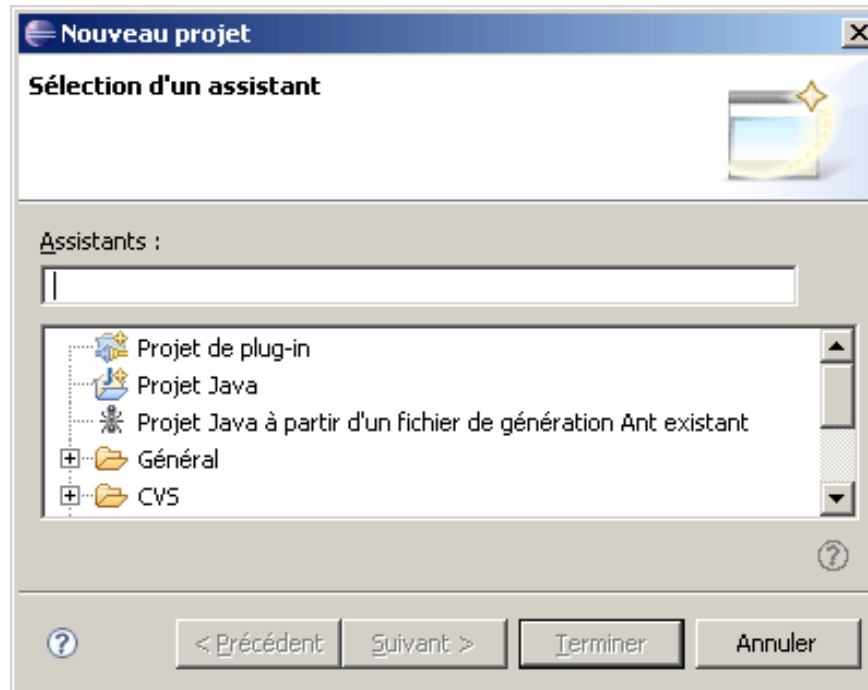
# Vues et éditeurs



# Assistants

## □ Assistant

- Facilite la saisie et/ou sélection d'information par des panneaux successifs
- Très nombreux et très utilisés



# L'espace de travail : Workspace

## □ Workspace

- Ressources : fichiers, répertoires, projets, etc.
- Espace de travail = un ou plusieurs projets

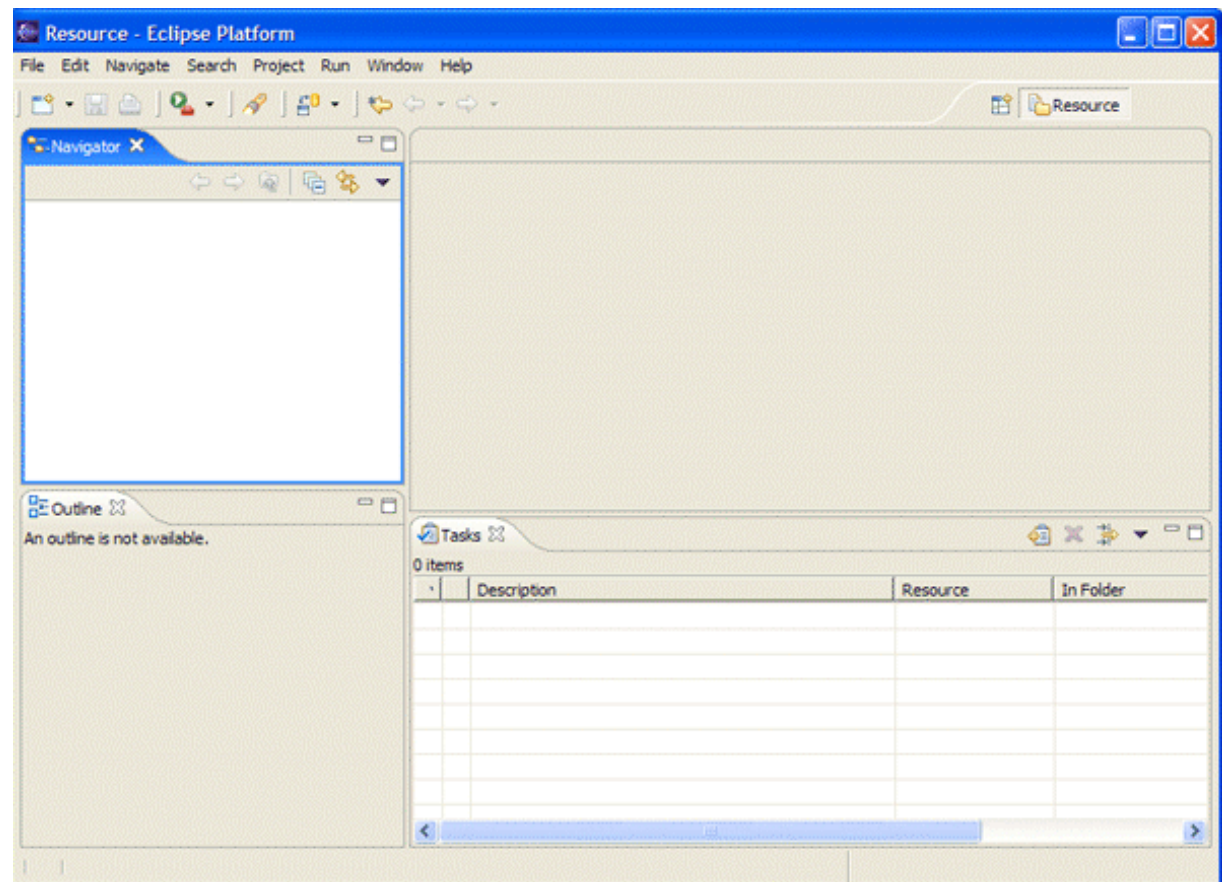
## □ Projet = partie du système de fichiers qui a une personnalité (définie par les plug-ins)

- Exemples : projet Java, site Web.

## □ Le workspace implémente un mécanisme d'historique locale (backup) pour tracer les changements des ressources

# Perspective Ressource

- ❑ Par défaut, cette perspective contient les fenêtres suivantes :
  - la vue "Navigateur" qui affiche les ressources (arborescence des fichiers) de l'espace de travail
  - un éditeur qui permet d'éditer une ressource sélectionnée dans la vue "Navigateur"
  - la vue "Structure" qui permet d'obtenir une arborescence présentant les grandes lignes de certaines ressources en cours de traitement
  - la vue "Tâches" qui affiche une liste de tâche à effectuer



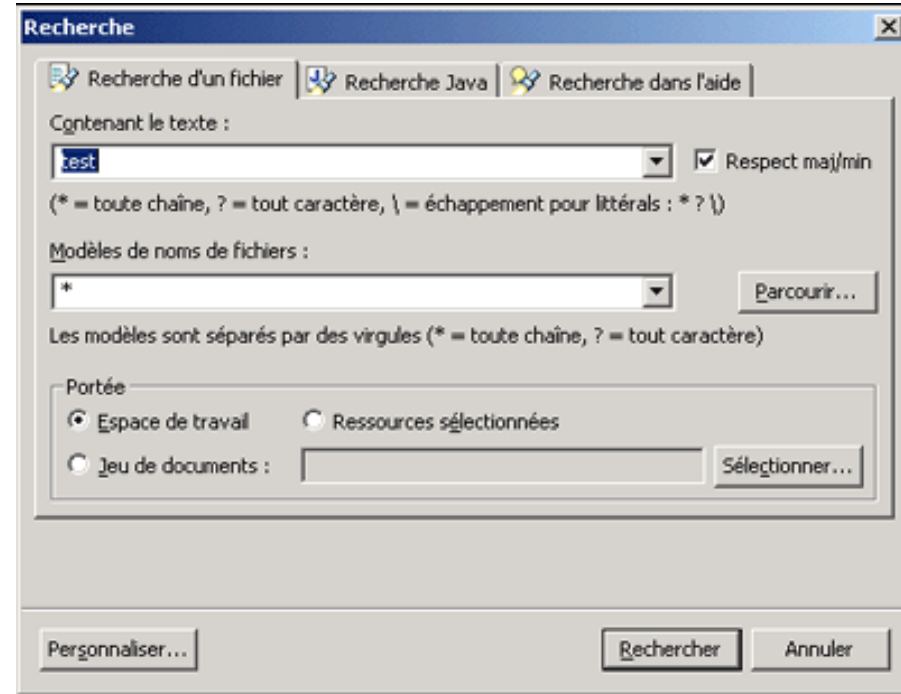
# Fonctions pratiques

## ❑ Recherche

- Dans tout l'espace
- Dans des fichiers
- Une recherche dédiée à Java

## ❑ Taches

- actions à réaliser
- erreurs de compilation à corriger
- points d'arrêt pour le débogage



C	I	Description	Ressource	Dans le dossier	Emplacement
✗		L'importation javax.ejb ne peut pas être résolue	MonPremierEJBBean.java	test_12EE/com/moi/ejb	ligne 3
✗		SessionBean ne peut pas être résolue ou ne corre...	MonPremierEJBBean.java	test_12EE/com/moi/ejb	ligne 10 dans ...
✗		Erreur de syntaxe sur le mot clé "else", "case", "d...	TestAssert1.java	test_perso	ligne 21 dans ...

Tâches Rechercher

- ❑ **F1 : aide contextuelle**
  - Dépend de la vue, l'éditeur, etc.
- ❑ **Dans un éditeur :**
  - CTRL + ESPACE => complétion

# Références

## ❑ Site Eclipse

- <http://www.eclipse.org/>

## ❑ Tutoriaux de JM Doudoux (le roi du screenshot !)

- [http://www.jmdoudoux.fr/accueil\\_java.htm#dejae](http://www.jmdoudoux.fr/accueil_java.htm#dejae)

## ❑ Divers supports de cours pour Eclipse (très bien aussi pour les screenshots) :

- <http://eclipse.developpez.com/cours/>