

Inhoud

Inhoud

## **Basishandleiding Eclipse**

Introductie

- 1 Starten en workspace kiezen
- 2 Instellingen
- 3 Basisfuncties
  - 3.1 Nieuw project, package, klasse maken
  - 3.2 Projecten en bestanden openen en sluiten
  - 3.3 Code bewerken
  - 3.4 Wijzigingen aanbrengen met refactoring
  - 3.5 Compileren en verwerken
  - 3.6 Uitleveren: documentatie en jars
  - 3.7 Resources toevoegen aan een project
  - 3.8 Diversen
- 4 De debugger
- 5 De Visual Editor
- 6 Testen met JUnit

# Basishandleiding Eclipse

## INTRODUCTIE

In deze handleiding is kort beschreven hoe u met behulp van Eclipse Java-programma's kunt schrijven.

De installatie van Eclipse wordt in dit document niet beschreven; hiervoor bestaat een apart installatiedocument.

In deze handleiding is verder alleen datgene opgenomen, dat voor alle gebruikers van Eclipse relevant is. Voor sommige cursussen zijn er daarnaast gespecialiseerde handleidingen, bijvoorbeeld voor het werken met de Visual Editor, voor het werken met het testframework JUnit en voor het werken met de XML-editor oXygen.

Deze handleiding is vrijwel geheel tekstueel; er zijn slechts enkele schermafdrukken opgenomen. In aanvulling op de handleiding komen er in de loop van 2008 filmpjes beschikbaar alsmede een FAQ waarin oplossingen worden aangegeven voor problemen.

In dit document wordt de notatie X | Y | Z ... gebruikt om aan te geven dat u eerst X moet kiezen, dan Y, dan Z ... X, Y en Z zijn daarin vaak menu's en submenu's, maar soms ook tabbladen of zelfs knoppen op tabbladen.

### 1 Starten en workspace kiezen

– Start Eclipse vanuit het menu Start als u het daaraan hebt toegevoegd (zie paragraaf 4.2 van de installatiehandleiding), of vanuit de installatiemap. Er is bij installatie niet vanzelf een pictogram op het bureaublad geplaatst.

– U krijgt een venster Workspace Launcher te zien; kies hierin een bestaande workspace of wijs (via de knop Browse) een nieuwe workspace aan.

#### Toelichting

Elk Java-programma is onderdeel van een *project*. Eclipse organiseert projecten in zogeheten *workspaces*. Een workspace is een map met een aantal projecten die alle instellingen met elkaar delen. Programma's die bij een bepaalde cursus horen, worden samengebracht in één workspace. Raadpleeg de cursusspecifieke informatie voor eventuele eisen aan die workspace (sommige cursussen zetten een workspace met bouwstenen klaar). Indien er geen bouwstenen zijn wijst u zelf een map als workspace aan.

– Vink eventueel ook het hokje 'Use this as default and do not ask again' aan (u kunt altijd vanuit Eclipse naar een andere workspace overschakelen, zie paragraaf 3.8).

– Klik op OK.

Mogelijke fout bij starten

Het kan voorkomen dat u na de installatie van Java, bij het starten van Eclipse een melding krijgt van de Java Virtual Machine Launcher dat de Java Runtime Environment niet is gevonden. Zie paragraaf 2 van de installatiehandleiding voor de installatie van Java en paragraaf 3 voor het instellen van de omgevingsvariabelen.

Na het aanwijzen van een nieuwe workspace start Eclipse op met het scherm van figuur 1 (de toelichting bij de verschillende knoppen is door ons toegevoegd).

– Klik op de knop Naar de ontwikkelomgeving.

Indien u een bestaande workspace heeft gekozen, ziet u meteen het hoofdvenster van Eclipse (zie paragraaf 3).



FIGUUR 1 Het welkomtscherm van Eclipse

## 2 Instellingen

In alle beschrijvingen en filmpjes is uitgegaan van bepaalde instellingen van Eclipse. We beschrijven hoe u deze instellingen kunt controleren en zonodig wijzigen.

– Kies uit het menu Window de optie Preferences; alle volgende aanwijzingen hebben betrekking op dat scherm.

Regelnummers

– Kies General | Editors | Text Editors.

Vink de optie Show line numbers aan en *klik op Apply* (vergeet dit laatste vooral niet!).

Spellingscontrole

– Kies General | Editors | Text Editors | Spelling.

	<p>Vink Enable spell checking uit en klik op Apply.</p>
Member sort order	<p>– Kies Java   Appearance   Members sort order: Selecteer rechts het item Static methods en verplaats dat met behulp van de knop Down naar beneden tot het onder Constructors staat en klik op Apply.</p>
Broncode en .class-bestanden apart	<p>– Kies Java   Build Path Kies onder Source and output folder de optie Folders. Vul bij Source folder name in: src Vul bij Output folder name in: classes. Klik op Apply. <i>Opmerking</i> In de bouwstenen en projecten van de cursus Objectgeoriënteerd programmeren in Java 1 staan de java-bestanden en de class-bestanden wel bij elkaar (het betreft daar hele kleine projecten).</p>
Nieuwe vormgeving voor code aanmaken	<p>– Kies Java   Code Style   Formatter. Klik op de knop New om een eigen standaard aan te maken voor de vormgeving van de code. U krijgt een venstertje New Code Formatter Profile. Vul een naam (<i>naam</i>) in en kies uit de lijst daaronder Java Conventions (we blijven in onze vormgeving van de code dicht bij deze conventies). Klik tot slot in dit deelvenster op OK.</p> <p>Er wordt nu een venster geopend Profile <i>naam</i>, met verschillende tabbladen.</p>
Inspringen	<p>– (Kies zonodig eerst Java   Code Style   Formatter; Edit profile <i>naam</i>). Wijzig op het (voorliggende) tabblad Indentation de Tab Policy in Spaces only, de Indentation size in 2 en vink in de lijst van hetgeen moet worden ingesprongen, ook Statements within 'switch'body aan. Klik ten slotte op Apply.</p>
Witregels	<p>– (Kies zonodig eerst Java   Code Style   Formatter; Edit profile <i>naam</i>). Wijzig op het tabblad Blank Lines in het veld Before import declaration de 1 in een 0; doe hetzelfde voor de velden Between import groups en Before field declarations. Klik dan weer op Apply.</p>
Control statements	<p>– (Kies zonodig eerst Java   Code Style   Formatter; Edit profile <i>naam</i>). Vink op het tabblad Control statements alle vier de opties onder General aan. U kunt nu op OK klikken, waardoor het venster Edit profile OPIJ sluit.</p>
Compiler compliance level	<p>– Kies Java   Compiler. Zet het Compiler compliance level op 6.0 en klik op Apply. Het kan zijn dat Eclipse dan vraagt of u een Full Rebuild wilt: kies dan Yes.</p>
Visual editor op apart tabblad	<p>– Kies Java   Visual Editor. Kies op het tabblad Appearance bovenaan bij Show the visual editor and source editor de mogelijkheid On separate notebook tabs. Klik op Apply.</p>
Default Layout manager null	<p>– Kies Java   Visual Editor   Swing. Plaats in de lijst die u dan te zien krijgt (Default Layout Manager) het vinkje bij null en klik op Apply.</p>

Always save dirty editors

– Kies Run/Debug | Launching.  
Kies op de eerste regel, bij Save required dirty editors before launching, de mogelijkheid Always. Klik op OK.

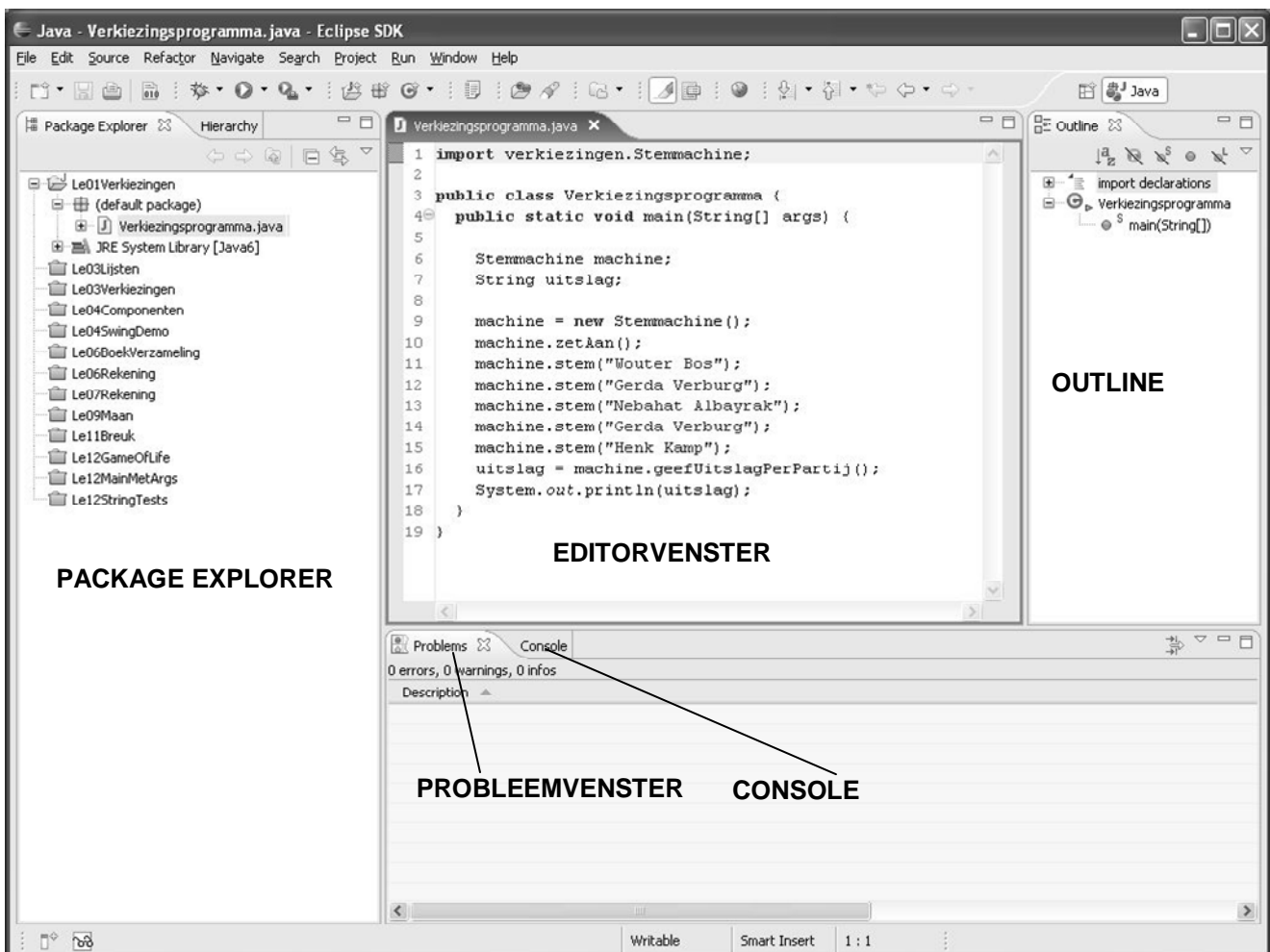
Java perspective tonen

Het hoofdvenster van Eclipse moet zo ingesteld worden dat het Java perspective getoond wordt. U kunt dit doen via Window | Show perspective | Java, of via de knop Show perspective rechtsboven in beeld. Als het Java perspective te zien is, is ook de (ingedrukte) knop Java perspective rechtsboven zichtbaar.

Views die zichtbaar moeten zijn

Verder moeten de volgende views zichtbaar zijn. U kunt deze zonodig selecteren via Window | Show view.  
– Links de *package explorer*. Een alternatief hiervoor is Navigator, maar deze view toont meer dan u nodig hebt (met name ook alle .class-bestanden).  
– Onder de tabbladen *Console* en *Problems*.  
– Rechts de *outline*.

Figuur 2 toont deze vensters:



FIGUUR 2 Hoofdvenster van Eclipse

### 3 Basisfuncties

#### 3.1 NIEUW PROJECT, PACKAGE, KLASSE MAKEN

- Nieuw project maken
- Kies File | New en dan Project (of Java Project, als die optie er is. U kunt de volgende stap dan overslaan.).
  - Kies Java | Java project en klik op Next (typ niets in het tekstvak bovenin).
  - Voer een projectnaam in en klik op Finish.
- Nieuw project maken uit bestaande sources
- Kies File | New en dan Project.
  - Kies Java | Java project en klik op Next (typ niets in het tekstvak bovenin).
  - Voer een projectnaam in.
  - Kies onder Contents de optie Create project from existing source
  - Klik op Browse, blader naar de map waar de sources staan en klik op OK.
  - Klik op Finish.
- NB: Dit werkt alleen als de sources *niet* al binnen de workspace staan
- Nieuwe package maken
- Klik rechts op het project waarbinnen de package moet komen en kies New | Package.
  - Vul de packagenaam in en klik op Finish.
  - U kunt een package ook tegelijk met een klasse creëren.
- Klasse maken in bestaande package
- Klik rechts op de package waarbinnen de klasse moet komen en kies New | Class; de packagenaam is al ingevuld.
  - Vul de naam van de klasse in.
  - Wijs eventueel een superklasse en/of te implementeren interfaces aan (niet voor de cursus Objectgeoriënteerd programmeren met Java 1).
  - Klik op Finish.
- Klasse maken in nieuwe package
- Klik rechts op het project waarbinnen de klasse moet komen en kies New | Class.
  - Vul de naam van een nieuwe package in.
  - Vul de naam van de klasse in.
  - Wijs eventueel een superklasse en/of te implementeren interfaces aan (niet voor de cursus Objectgeoriënteerd programmeren met Java 1).
  - Klik op Finish. De klasse wordt gecreëerd binnen de nieuwe package.

#### 3.2 PROJECTEN EN BESTANDEN OPENEN EN SLUITEN

- Project openen
- Klik in de Package Explorer rechts op de projectnaam en selecteer *Open Project* of dubbelklik op de projectnaam
  - Onderdelen worden zichtbaar (onzichtbaar) gemaakt via klikken op + (-).
- Openen bestanden
- Klik om een bestand binnen een project te openen rechts op de bestandsnaam en kies Open of dubbelklik op de bestandsnaam. Als het bestand niet op de juiste manier wordt geopend, kies dan Open with... in plaats van Open en kies de juiste editor.
- Project sluiten
- Klik in de Package Explorer rechts op de naam van het project en kies voor Close Project.

### 3.3 CODE BEWERKEN

- Code wordt bewerkt in het editorvenster met behulp van de Java editor.
- Compilatiefouten herstellen
  - Foutmeldingen van de compiler worden in de programmatekst aangegeven.
  - Indien in de kantlijn behalve een rood kruisje ook een lampje staat, dan kunt u daarop klikken (of er rechts op klikken en kiezen voor de optie Quick Fix); er verschijnt dan een menu met suggesties voor reparatie.
- Importopdrachten
  - Gebruik klassen uit andere packages zonder deze vooraf te importeren. Klik in de kantlijn op het lampje bij de foutmelding (of klik rechts en kies de optie Quick Fix) en selecteer de importopdracht.
- Code completion
  - Indien u na de naam van een klasse of een object een punt intypt en dan even wacht, verschijnt een venster met alle toepasselijke attributen en methoden. Typen van de eerste letter(s) perkt de keuzemogelijkheden overeenkomstig in.
  - CTRL + spatie toont alle zichtbare namen. Typen van de eerste letter(s) perkt de keuzemogelijkheden overeenkomstig in.
- Code tot commentaar maken
  - Selecteer het deel van de code dat u tijdelijk wilt verwijderen en typ CTRL + /. Het geselecteerde deel wordt tot commentaar.
  - Later kunt u dit commentaar selecteren en opnieuw CTRL + / intypen; de commentaartekens worden dan weer verwijderd.

### 3.4 WIJZIGINGEN AANBRENGEN MET REFACTORING

Met behulp van refactoring kunnen in één keer samenhangende wijzigingen in de code worden aangebracht. We noemen hier slechts enkele basale mogelijkheden.

- Naam wijzigen
  - In Eclipse kan elke naam gewijzigd worden. Voor de namen van projecten, packages, klassen, attributen en methoden kan dit als volgt:
    - Rechtsklik in de Package Explorer op het element waarvan u de naam wilt wijzigen en kies Refactor | Rename.
    - Voer in het getoonde venster de nieuwe naam in.
    - Klik altijd 'Update references' aan en desgewenst ook 'Update similarly named variables and methods' (om bijvoorbeeld meteen ook de get-methode van een attribuut te hernoemen) en 'Update textual occurrences in comments and strings'.
    - Klik op Next als er een preview noodzakelijk is of als u die wenst.
    - Klik op Finish.
- Namen in code wijzigen
  - Namen kunnen ook vanuit de codeditor gewijzigd worden.
    - Klik in de code op de naam die u wilt wijzigen, zodat de cursor er op staat.
    - Klik rechts en kies Refactor.
    - Wijzig de naam en geef enter.
    - Volg de aanwijzingen.

- Klassen verplaatsen
- Rechtsklik in de Package Explorer op de klasse die u wilt verplaatsen en kies Refactor | Move.
  - Kies in het venster dat u te zien krijgt de package waarheen u de klasse wilt verplaatsen, of klik Create New Package en maak een nieuwe package.
  - U kunt als bestemming niet alleen packages binnen hetzelfde project kiezen, maar ook packages binnen een ander project in dezelfde workspace, mits dat project geopend is.

### 3.5 COMPILEREN EN VERWERKEN

- Compileren en verwerken applicatie
- Klik in de Package Explorer binnen een geopend project rechts op de klasse die de methode main bevat, ga met de muiswijzer naar Run as... en selecteer vervolgens Java Application
  - Eventuele foutmeldingen van de compiler verschijnen in het tabblad Problems midden onder.
  - Standaard uitvoer verschijnt in het tabblad Console midden onder.
  - Wilt u hetzelfde programma (eventueel na wijziging) nogmaals laten verwerken, dan kunt u ook gebruikmaken van de groene knop op de knoppenbalk die onder de menubalk zit. Naast die knop zit een menuutje dat de programma's toont die onlangs zijn verwerkt.

- Compileren en verwerken applet in applet viewer
- Klik in de Package Explorer binnen een geopend project rechts op de applet-klasse, ga met de muiswijzer naar Run as... en selecteer vervolgens Java Applet. Zie verder hierboven.

- Parameters meegeven aan main
- Rechtsklik op de klasse met de methode main waar u parameters aan mee wilt geven en selecteer Run as | Open Run Dialog.
  - Controleer in het getoonde venster op het tabblad 'Main' dat de juiste klasse is aangegeven; pas deze zonedig aan.
  - Kies het tabblad '(x) = Arguments'.
  - Vul in het veld Program arguments de waarden van de actuele parameters in, gescheiden door spaties.
  - Klik op Run.

### 3.6 UITLEVEREN: DOCUMENTATIE EN JARS

- Javadoc genereren
- Kies uit het menu Project (van de menubalk!) de optie Generate Javadoc.
  - Controleer bovenin het venster dat u te zien krijgt de verwijzing naar het javadoc-commando en pas dit zonedig aan; het moet gelijk zijn aan <JAVA\_HOME>/bin/javadoc.exe, waarbij voor <JAVA\_HOME> de map moet worden ingevuld waar u Java geïnstalleerd hebt (bijvoorbeeld C:\Program Files\Java\jdk1.6.0\_04).
  - Kies voor welke elementen er javadoc gegenereerd moet worden. Links kunt u kiezen uit alle geopende projecten. Als u voor slechts een deel van het project javadoc wilt genereren, kunt u een project openklikken en dan voor packages kiezen. Rechts verschijnen dan de individuele klassen; ook die kunt u aan- of uitvinken.
  - Selecteer tot welk niveau van toegang javadoc gegenereerd moet worden (public indien u documentatie wilt voor programmeurs die de klasse gebruiken).
  - Selecteer een bestemming voor de HTML-pagina's met de javadoc. U kunt hier ook gewoon Use standard doclet aangevinkt laten staan. In dat geval vraagt Eclipse daarna 'Do you want to update the Javadoc location



for *projectnaam* with the chosen destination folder *automatisch ingevulde folder?*'. Klik dan op Yes to all.

- Klik op Finish.
- Tijdens het genereren van de documentatie verschijnt er een lange serie meldingen in de Console. Bekijk eventuele waarschuwingen. Ze betreffen meestal fouten in de javadoc-syntaxis.

Javadoc documentatie bekijken

Na het genereren van de documentatie bij de klasse *klasse* zijn er een aantal mappen met 'doc' in de naam aan het project toegevoegd.

- Zoek in deze mappen naar het bestand *klasse.html*, rechtsklik op dit bestand en kies Open With | Web browser; u bekijkt de documentatie bij *klasse* dan in Eclipse.
- U kunt ook naar de map bladeren die u heeft opgegeven en daar het bestand *index.html* openen; u krijgt dan een scherm met dezelfde layout als de API Specification.

JAR-file maken

- Kies File | Export
- Kies in het volgende venster Java | JAR file (doe niets aan het bovenste tekstveld waarin 'type filter text' staat) en klik op Next.
- Kies welke projecten er in de JAR opgenomen moeten worden. Links kunt u kiezen uit alle geopende projecten. Als u slechts een deel van het project wilt opnemen, kunt u een project openklikken en dan individuele packages kiezen. Rechts verschijnen dan de individuele klassen; ook die kunt u aan- of uitvinken.
- Kies eventueel welk type bestanden opgenomen moeten worden (standaard alleen .class-files en bestanden waarnaar vanuit de .class-files verwezen wordt, zoals plaatjes).
- Klik op Finish. Als er fouten in het project zaten, dan wordt dit vermeld; u moet die dan herstellen en daarna opnieuw een JAR maken.

### 3.7 RESOURCES TOEVOEGEN AAN EEN PROJECT

Toevoegen van JAR

- Klik in de package explorer rechts op het project waaraan u de JAR wilt toevoegen en kies Build Path | Configure Build Path.
- Kies het tabblad Libraries.
- Kies Add External JARS
- Blader naar de jar die u wilt toevoegen en klik op Openen.
- Klik op OK.

Deelen van ander project gebruiken

U kunt in project A klassen uit een project B van dezelfde workspace hergebruiken door project B aan het Build path van A toe te voegen.

- Open zowel Project A als project B.
- Klik in de package explorer rechts op project A en kies Build Path | Configure Build Path.
- Kies het tabblad Projects.
- Kies Add
- Kies project B uit de lijst van geopende projecten
- Klik op OK en nog een keer op OK.

### 3.8 DIVERSEN

Wisselen van workspace

- Kies File | Switch Workspace
- Kies indien mogelijk uit het lijstje rechts meteen de juiste workspace of kies Other als die er niet bijstaat.

- Als de workspace niet in het lijstje stond, krijgt u de Workspace Launcher te zien (zie paragraaf 1). Kies daarin de juiste workspace.

– Eclipse start opnieuw op in de gekozen workspace.
- Projectbestanden al aanwezig

Als u de bestanden voor een project als zipfile van Studienet haalt, gaat u als volgt te werk:

  - pak de bestanden uit naar de workspace
  - kies File | Import
  - kies General | Existing projects into Workspace en klik Next
  - select Root directory: browse naar workspace
  - vink de te importeren projecten aan
  - klik op Finish.
- Code genereren voor oudere JVM

  - Rechtsklik in de package explorer op het betreffende project en kies Properties.
  - Kies in het linker menu van het getoonde venster ‘Java compiler’.
  - Vink ‘Enable project specific settings’ aan en selecteer het gewenste level.
  - Klik op OK. Als er al klassen in het project waren opgenomen is een rebuild nodig; bevestig dit door op Yes te klikken. Er wordt een oudere versie gebruikt; u kunt dus foutmeldingen krijgen!

#### 4 De debugger

Met behulp van de debugger kunnen fouten in een programma worden opgespoord. Met de debugger kan de verwerking van het programma op elk gewenst moment onderbroken worden en kunnen de waarden geïnspecteerd worden die de verschillende zichtbare variabelen op dat moment hebben. Vervolgens kan de code desgewenst stap voor stap doorlopen worden. Deze paragraaf geeft enkele basismogelijkheden weer van de debugger van Eclipse. We tonen deze in de volgorde waarin u ze tegenkomt als u een programma gaat debuggen.

Er zijn vaak verschillende mogelijkheden waarop u een actie kunt uitvoeren, we noemen de handigste.

- Plaatsen van een breakpoint

Een *breakpoint* markeert een regel waar de verwerking van de code onderbroken moet worden zodra deze regel bereikt is, net voor de verwerking daarvan. Het plaatsen van een breakpoint gaat als volgt:

  - Klik rechts op het regelnummer van de regel waar u het breakpoint wilt plaatsen en kies Toggle Breakpoint

Het breakpoint wordt gemarkeerd door een gevuld blauw bolletje helemaal links in de betreffende regel.
- Disablen van een breakpoint

  - Klik rechts op het breakpoint (het bolletje) en kies Disable Breakpoint. Het breakpoint wordt nu gemarkeerd door een leeg blauw bolletje.
- Enablen van een breakpoint

  - Klik rechts op het breakpoint (het bolletje) en kies Enable Breakpoint. Het breakpoint wordt weer gemarkeerd door een gevuld blauw bolletje.
- Verwijderen van een breakpoint

  - Klik rechts op het breakpoint en kies Toggle Breakpoint.

Het plaatsen, disablen / enablen en verwijderen van breakpoints kan ook via het menu Run; de cursor moet dan in de code editor op de juiste regel staan.

Verwerken van een klasse in de debugger

– Klik in de Package Explorer binnen een geopend project rechts op de klasse die de methode main bevat, ga met de muiswijzer naar Debug as... en selecteer vervolgens Java Application.

In de code-editor kunt u zien tot welke regel de verwerking van het programma gevorderd is: we zullen deze regel verder aanduiden als het *verwerkingspunt*.

Op het moment dat het programma het eerste breakpoint bereikt, verschijnt er een venster met titel 'Confirm Perspective Switch',  
– Kies Yes.

Eclipse schakelt nu om van het (normale) *Java perspective* (figuur 2) naar het *Debug perspective* (zie figuur 3).  
In het debug perspective kunnen ook weer verschillende deelvensters zichtbaar zijn. Zorg dat in elk geval de vensters zichtbaar zijn getoond in figuur 3.

Debug View

– Maak zonodig de Debug View zichtbaar met Window | Show View | Debug.  
Dit venster toont de call stack tot aan het verwerkingspunt (dus alle methoden die op dat moment zijn aangeroepen en nog niet zijn teruggekeerd).

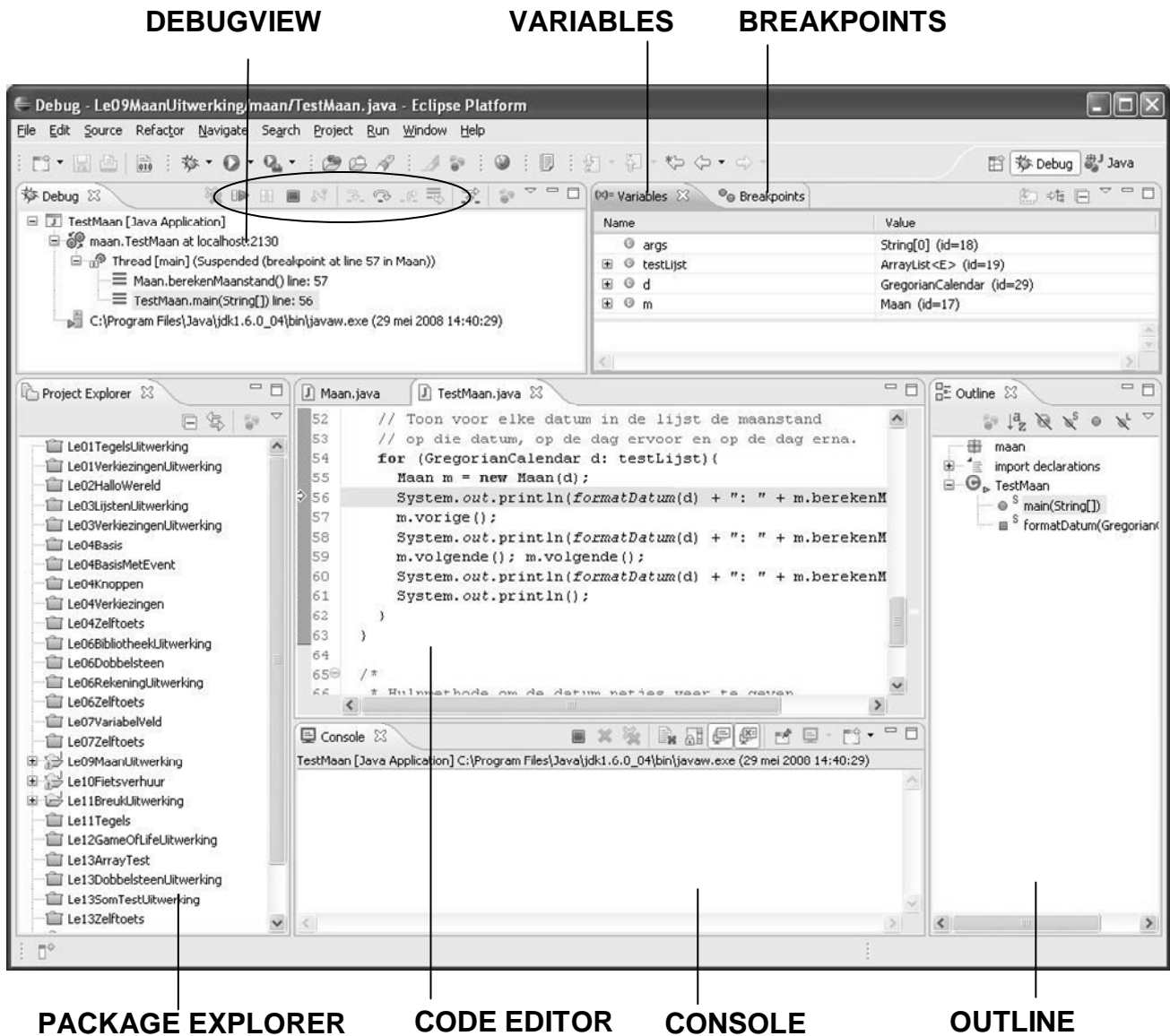
Breakpoints View

– Maak zonodig de Breakpoints View zichtbaar met Window | Show View | Breakpoints.  
Dit venster geeft een overzicht van de breakpoints die gezet zijn. U kunt breakpoints enablen of disablen door middel van een vinkje. Door rechts op het breakpoint te klikken verschijnt een menu; hiermee kunt u onder meer het breakpoint verwijderen (of zelfs alle breakpoints verwijderen).

Variables View

– Maak zonodig de Variables View zichtbaar met Window | Show View | Variables.  
Dit venster toont de inhoud van alle variabelen die zichtbaar zijn op het verwerkingspunt.  
– De waarden van de onderdelen van variabelen van een referentietype worden zichtbaar door (herhaaldelijk) op het ' + teken' te klikken.

Vanuit het Java Perspective kunt u het Debug Perspective altijd openen via Window | Open Perspective | Other | Debug of met de knop Debug rechts op de knoppenbalk.



FIGUUR 3 Views in Debug perspective

U kunt nu vanaf het breakpoint op verschillende manieren stap voor stap door de code lopen en bij elke stap de waarde van de variabelen inspecteren.

Step Over

– Kies in de menubalk bovenin Run | Step Over om naar de volgende opdracht in dezelfde methode te gaan. Een aanroep van een methode of constructor geldt dan dus als een opdracht, die in zijn geheel wordt verwerkt voor de verwerking weer stopt. U kunt ook de betreffende knop gebruiken op de knoppenbalk in het Debug Perspective (binnen de ellips in figuur 3).

Step Into

– Kies Run | Step Into (of de bijbehorende knop op de knoppenbalk) als de eerstvolgende opdracht in de code een methodeaanroep of aanroep van een constructor is en u ook de opdrachten daarvan

stapsgewijs wilt doorlopen. De verwerking stopt dan meteen na de aanroep.

- Step Return – Kies Run | Step Return (of de bijbehorende knop op de knoppenbalk) indien u vanuit de code van een methode *m* in een keer terug wilt naar de aanroeper van *m*.
- Waarde van expressies  
Inspecteer de waarde van een expressie als volgt:
  - Selecteer in de code een expressie waarvan de waarde bekend is op het verwerkingspunt.
  - Kies Run | Inspect (of klik rechts op de geselecteerde expressie) en kies voor Inspect).Er verschijnt een panel met de waarde van de expressie.
- Resume – Kies Run | Resume (of de bijbehorende knop op de knoppenbalk) om het programma te verwerken tot het volgende breakpoint.
- Terminate – Kies Run | Terminate (of de bijbehorende knop op de knoppenbalk) om de verwerking van het programma te beëindigen.
- Terug naar Java Perspective – Kies Window | Open Perspective | Java of de knop Java rechts op de knoppenbalk om terug te keren naar het Java Perspective.

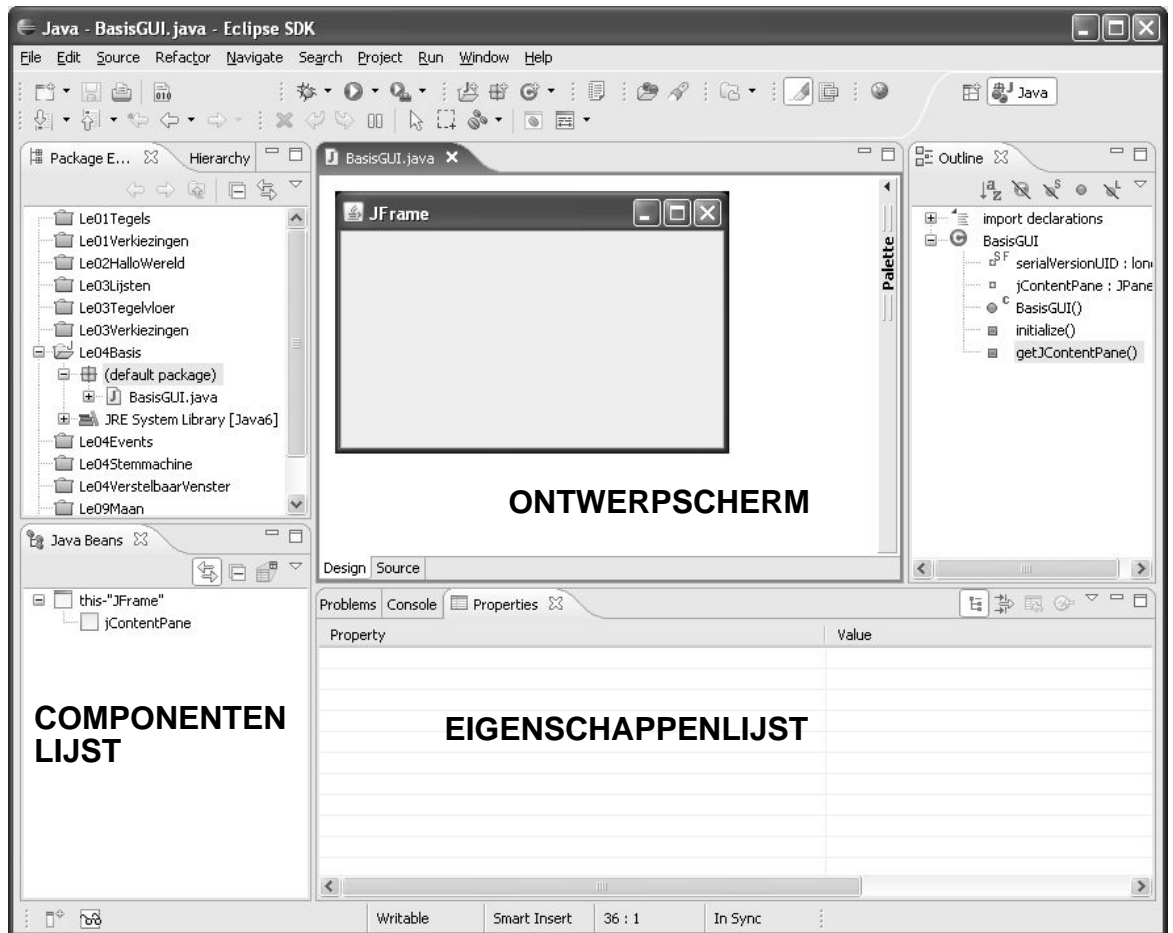
Tot slot een *tip*:

De drop down list bij de knop met het torretje bevat de Java programma's die het meest recent verwerkt zijn. Door een programma uit de lijst te selecteren kunt u deze eenvoudig (opnieuw) verwerken met de debugger.

## 5 De Visual Editor

Voor het maken van grafische gebruikersinterfaces beschikt Eclipse over een Visual Editor. De programmeur kan met behulp daarvan een interface als het ware op het scherm 'tekenen'; de bijbehorende code wordt automatisch gegenereerd.

- Views  
Bij gebruik van de Visual Editor moeten, naast de views getoond in figuur 2, ook de volgende views zichtbaar zijn (zie figuur 4):
  - In het midden het ontwerpscherm (tabblad of apart venster Design).
  - Links onderin de componentenlijst (Java Beans). Is deze niet zichtbaar, dan kunt u deze vinden via Window | Show view | Other | Java | Java beans.
  - Beneden een extra tabblad Properties (de eigenschappenlijst).
- Openen van een klasse in de Visual Editor
  - Klik rechts op de bestandsnaam, kies Open with... en vervolgens Visual Editor.
  - Indien het bestand eerder geopend was met de Visual Editor, wordt het bij dubbelklikken vanzelf opnieuw daarin geopend.



FIGUUR 4 Views bij gebruik van de Visual Editor

Weergave GUI's die niet met de VE van Eclipse zijn gemaakt

Alleen GUI-klassen die met behulp van de Visual Editor van Eclipse gemaakt zijn, worden gegarandeerd goed weergegeven. De VE probeert GUI-klassen die op een andere manier zijn gemaakt, ook goed weer te geven, mits de VE de juiste initialisatiemethode kan identificeren. U kunt deze als volgt aanwijzen:

- Kies Window | Preferences | Java | Visual Editor
- Ga naar het tabblad Pattern Styles
- Klik op Add.
- Voer de naam van de initialisatiemethode in
- Klik op OK en dan op Apply (of nogmaals OK).

Merk op dat jbInit al bij de methodenamen voorkomt; Eclipse kan dus overweg met GUI-klassen die gemaakt zijn met JBuilder.

Nieuwe Visual Class maken

- Klik rechts op het project of de package waarbinnen de klasse moet komen en kies New | Visual Class.
- Mocht Visual Class niet in de lijst staan, kies dan Other in plaats van Visual Class, klik in het venster dat u te zien krijgt op het plusje naast Java, kies Visual Class en klik op Next.
- Vul eventueel de packagenaam in.
- Vul de naam van de nieuwe klasse in.

- Selecteer in het venster Style het type grafische klasse dat u wilt maken (bijvoorbeeld Swing | Frame voor een JFrame; Swing | Applet voor een JApplet, Swing | Panel voor een JPanel enzovoort)
  - Klik op Finish.
  
- Hoofdvenster sluiten
  - Applicaties worden niet standaard gesloten als de frame wordt gesloten. Er blijft dan een loos proces draaien. Doe daarom voor JFrames het volgende, tenzij u zeker weet dat u bij het sluiten van de frame *niet* de hele applicatie wilt sluiten.
  - Ga naar de view Design.
  - Selecteer het gehele venster (klik in de titelbalk).
  - Geef in tabblad Properties aan defaultCloseOperation de waarde Exit.
  
- Layout managers
  - De standaard layout-manager van een grafische component is niet null, maar bijvoorbeeld BorderLayout (voor JFrame en JApplet) of GridBagLayout (JPanel). Wilt u geen layout manager of een andere, dan moet u dit wijzigen vóór u componenten toevoegt:
  - Klik binnen het venster rechts en kies Set Layout | null.
  
- Componenten toevoegen
  - Klik op de balk Palette aan de rechterkant van het ontwerpscherm.
  - Klik zonodig op de categorie Swing Components (andere categorieën zijn Swing Containers, Swing Menus en AWT Controls).
  - Klik in de palette op de gewenste component.
  - Klik vervolgens in de contentPane op de plek waar u de component wilt plaatsen en maak een rechthoek van de gewenste afmetingen.
  - Voer in het getoonde scherm 'name' een naam in voor de component en klik op OK.
  - Gebruik de eigenschappenlijst om de component de gewenste eigenschappen te geven.
  - De tekst van componenten die een setText methode hebben, zoals labels, knoppen en tekstvelden, kan ook worden ingevoerd door in het ontwerpscherm rechts te klikken op de component en vervolgens Set text te kiezen.
  
- Componenten op elkaar afstemmen
  - Selecteer op het ontwerpscherm de componenten die u wilt uitlijnen of waarvan u de afmetingen op elkaar af wilt stemmen.
  - NB: alle componenten worden aangepast aan de component die het *laatst* geselecteerd is.
  - Rechtsklik en kies voor Customize Layout.
  - U krijgt een schermje waarin u kunt kiezen voor verschillende vormen van uitlijnen (eerste twee kolommen), voor het gelijkmaken van breedte, hoogte of beide (derde kolom) of voor het verdelen van de componenten over de ruimte (laatste kolom).
  
- Event handler toevoegen
  - Rechtsklik in het ontwerpscherm op de component waaraan u een event handler toe wilt voegen, kies events en vervolgens de event die u wilt toevoegen. Als deze niet in het menu staat, kies dan Add events... en selecteer de gewenste event uit het menu in het venster dat u dan krijgt. Voor het vervolg nemen we aan dat deze component de naam xComp draagt.
  - Voeg aan de broncode een private methode toe die de event afhandelt. Voor het vervolg nemen we aan dat deze methode de naam xCompAction draagt.
  - Zoek in de gegenereerde code binnen de methode getXComp()de volgende regel (op de stippeltjes staat de naam van de event).

```
System.out.println("...()"); // TODO Auto-generated Event stub  
...
```

en vervang deze door een aanroep naar `xCompAction()`.

Klasse verwerken

– Voeg zonodig aan de visuele klasse `XFrame` een methode `main` toe als volgt:

```
public static void main(String[] args) {  
    XFrame gui = new XFrame();  
    gui.setVisible(true);  
}
```

– Verwerk de klasse op de normale manier als applicatie (zie paragraaf 1 van deze bijlage).

Codevolgorde

Wij raden u aan om op de volgende manier elementen toe te voegen aan de gegenereerde code:

- plaats uw eigen attributen meteen na de kop van de klasse
- plaats de private methode `mijnInit` meteen na de constructor
- plaats de event handlers (private methoden `compAction`) na `mijnInit` of, indien er geen `mijnInit` is, na de constructor
- plaats andere private hulpmethoden na de event handlers
- plaats de methode `main` helemaal achteraan.

De volgorde van de code binnen een `JFrame`-klasse wordt dan als volgt:

- eigen attributen
- gegenereerde attributen (`serialVersionUID`, `jContentPane` en de toegevoegde componenten)
- gegenereerde constructor (waar eventueel een aanroep van `mijnInit` aan is toegevoegd)
- `mijnInit`
- methoden `compAction`
- private hulpmethoden
- gegenereerde methode `initialize`
- gegenereerde methoden `getJContentPane` en `getComp`
- methode `main`.

## 6 Testen met JUnit

JUnit is een unit testing framework voor Java, dat gebruikt kan worden voor het testen van afzonderlijke klassen. Bij iedere klasse wordt een testklasse gemaakt, die verwerkt kan worden elke keer dat de klasse is gewijzigd. JUnit is geen standaard onderdeel van de JDK en moet dus apart worden toegevoegd. De library is wel aanwezig in de door ons uitgeleverde Eclipse-versie en hoeft dus niet apart te worden opgehaald.

Eclipse biedt ondersteuning bij het maken van een testklasse en het uitvoeren van de tests.

JUnit library toevoegen

Om binnen een project gebruik te maken van JUnit, moet de juiste library aan dat project worden toegevoegd.

- Rechtsklik in de Package Explorer op het project waaraan u de JUnit-library toe wilt voegen en kies `Build Path | Add libraries`.



- JUnit testklasse maken
- Selecteer JUnit en klik op Next.
  - Selecteer als JUnit-library version JUnit 4.
  - Klik op Finish.
- JUnit testklasse maken
- Met behulp van Eclipse kan een skelet gemaakt worden voor een testklasse (de code van de testmethoden dient u uiteraard zelf in te vullen).
- Rechtsklik in de Package Explorer op de klassen die u wilt testen en kies New | JUnit Test Case.
  - Kies bovenin het venster dat u te zien krijgt voor New JUnit 4 test.
  - Vul een naam in voor de testklasse.
  - Klik op Next.
  - Vink de methode(n) aan die u getest wilt hebben.
  - Klik op Finish.
- JUnit test uitvoeren
- Rechtsklik op de testklasse die verwerkt moet worden en kies Run As | JUnit Test.
  - Links verschijnt een nieuw tabblad JUnit test.
  - De balk bovenin is groen als alle tests slagen; rood als dat niet het geval is.
  - In het bovenste deelvenster kunt u zien welke tests zijn uitgevoerd en wat het resultaat was (vinkje bij succes; kruisje bij falen).
  - In het onderste deelvenster staan de opgetreden fouten; net als bij gewone exceptions kunt u naar de betreffende regel gaan door dubbel te klikken op een regel die begint met at... en een regelnummer vermeldt.