

Verktyg och Utvecklingsmiljö

Joachim von Hacht

(mindre tillägg av Daniel Wetterbro)

Verktyg

Modern programutveckling innebär att man måste behärska ett antal verktyg

- Editorer
- Kompilatorer
- Avlusare (debugger)
- Versionhantering (kommer i projektkurs)
- Testverktyg (kommer i projektkurs)
- Verktyg för analys av kodkomplexitet, kodkvalité, kodeffektivitet, m.m.
- ...

Vi skall titta på Eclipse, Findbugs och STAN

Eclipse

En (ett?) IDE (integrated development environment)

Ett mycket avancerat ramverk. Kan användas för många olika sorters programmering (och många olika språk)

- Uppbyggt med "plugins"
- Finns massor av plugins, enkelt att installera direkt från nätet

Innehåller editor, debugger, m.m.

Ger mycket hjälp med trivial kod, fixa irriterande småfel, struktur, ..., måste dock **tänka själv!**



Eclipse Versioner

Finns tyvärr väldigt många, förvirrande... En gång varje år släpps en s.k. “simultaneous release” som bildar bas för andra versioner som inriktar sig på olika områden. En sådan release namngiven efter en astronomisk företeelse eller person, senaste heter Eclipse Luna:



Två uppdateringar (SR1 och SR2) görs också varje år vilket ger tre releaser per år, se http://wiki.eclipse.org/Simultaneous_Release

För att få den senaste och bättre prestanda, hämta en egen version istället för att köra den som är förinstallerad på Chalmers. Ingen installation krävs, bara att packa upp, i mappen eclipse, kör filen eclipse.

<http://www.eclipse.org/downloads/> (välj IDE for Java Developers)

SE UPP

Första gången du startar Eclipse frågas **eventuellt** efter ett **workspace**

- Det kommer att skapas en mapp "workspace" i din hemkatalog
- Svara ja, kryssa för "Inte fråga igen..."
- Du skall ha **exakt ett** workspace
- Du skall **aldrig spara** något i workspace, använd en kursmapp (~/TDA550 eller dyl)
- Om Eclipse får problem, ta bort workspace-mappen, ger en ny fräsch installation

Eclipse Arbetsflöde

1. Skapa projektkatalog, projektkatalogen skall innehålla alla .java-filer och övriga resurser programmet behöver (ikoner, bilder, ljud, ...)
2. Skapa ett Eclipse projekt med hjälp av t.ex. menyn File->New.
För att inte lägga projektet i ditt workspace, klicka ur "Use default location". Du behöver då själv skapa en ny mapp för projektet och ange sökvägen i "Location", använd t.ex. "Browse..."-knappen.

Starkt rekommenderat!

3. Lägg till paket, klasser, bibliotek m.m. m.h.a. Eclipse
4. Koda, kör, avlusa och faktorisera m.h.a. Eclipse
5. if(!ok) goto 3
6. Skapa jar-fil (slutgiltiga programmet) m.h.a Eclipse

Filstruktur för Projekt

Eclipse-projektet skapar en filstruktur i projektkatalogen

```
myproj.ep      // Project dir
|
|--- bin      // *.class-files. Created by Eclipse at compile
|   |--- se
|       |--chalmers
|
|--- src      // Package hierarchy for *.java-files
|   |---se    // Top level package
|       |--chalmers
```

Eclipse Projekt

Projekt, ett internt begrepp för Eclipse

- **Inte** samma som projektkatalog
- Kan ta bort projekt i Eclipse utan att radera kod (om du vill kan du ta bort koden också)
- Kan importera projekt
- Kan skapa projekt utifrån "lösa" Java-filer
- Kan ha flera projekt öppna samtidigt

För att Eclipse skall acceptera ett projekt måste filerna **.project** och **.classpath** finnas i projektkatalogen, skapa nytt (ev utifrån lösa filer)

En gång till: Inte spara projektet i workspace!

Eclipse Användargränssnitt

Finns väldigt många fönster... några begrepp...

View, ett internt fönster, t.ex. Navigator (filhanteringsfönster)

Editor, ett fönster där man kan arbeta med text

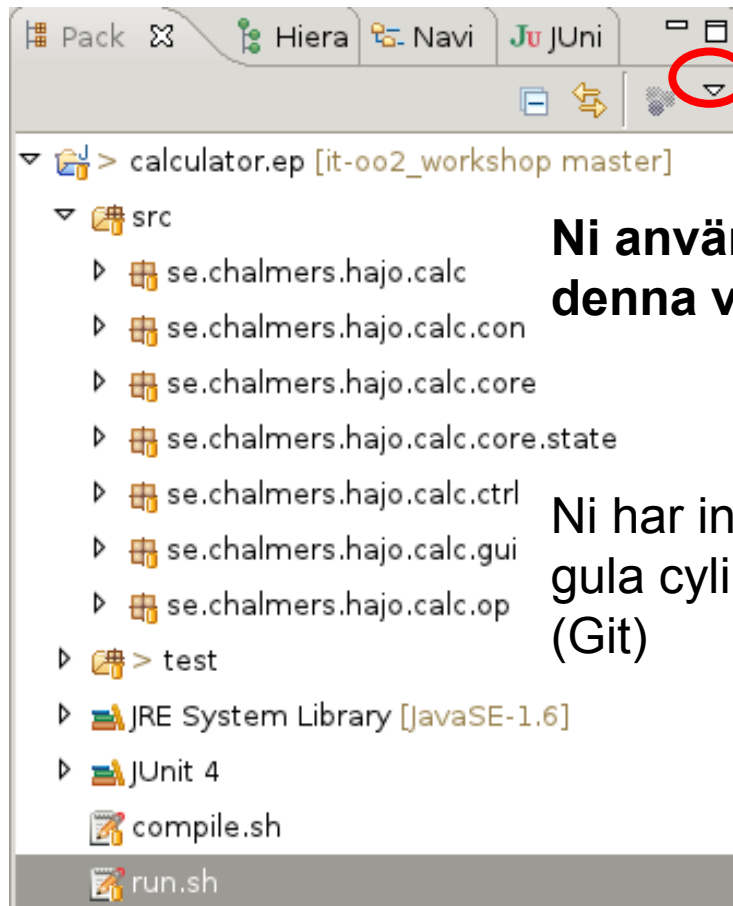
Perspective, ett visst arrangemang (layout) av ett visst antal views och editors

- Kan växla perspektiv (förvirrande för nybörjaren, fönster försvinner/dyker upp byter plats)
- **Problem: Markera perspektivet Java (Ikon långt upp höger, högerklick Reset)**

Det ovan finns att välja under Window-menyn

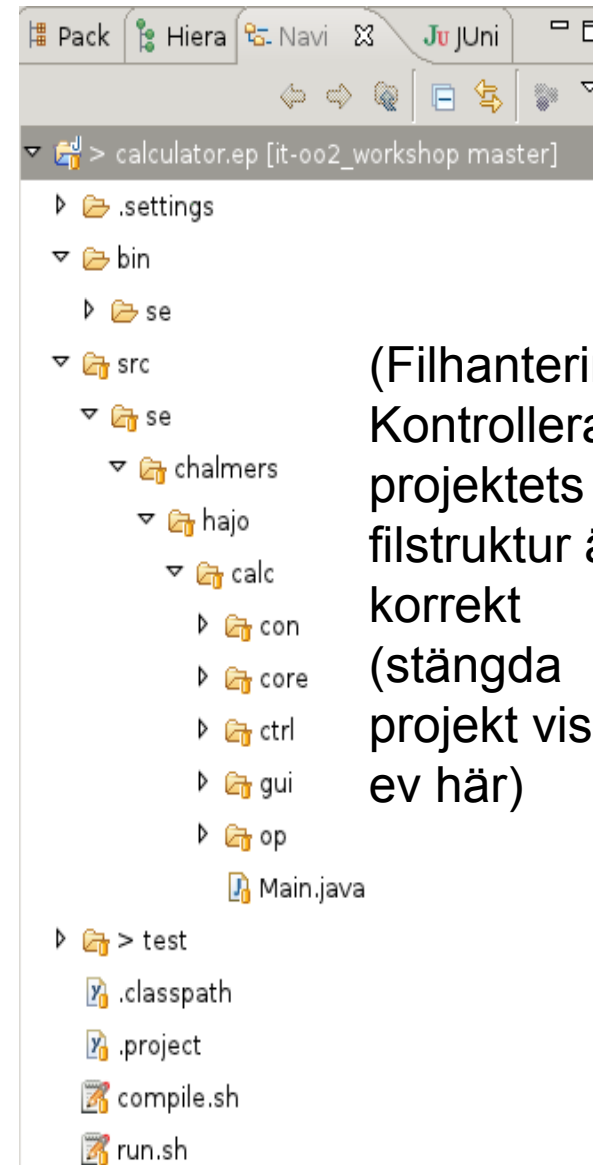
Olika Vyer av Projekt

Möjligt att modifiera vad som visas, **View Menu**, flat, hierarchical



Ni använder denna vy

Ni har inte små gula cylindrar (Git)



(Filhantering)
Kontrollera att projektets filstruktur är korrekt (stängda projekt visas ev här)

Eclipse Varningar och Felmeddelanden

Eclipse kompilerar kontinuerligt i bakgrunden

Varningar och fel visas i marginalerna

- Vänster: Ikoner, glödlampor m.m.
- Höger: Var i filen, röd(a) markering(ar)

Peka på ikoner för att få tips, klicka och välj ändringsförslag
(Eclipse föreslår,.. dock inte alltid optimalt)

Ibland "hänger" felmeddelandena kvar, **spara så försvinner de!**

- Kan också hjälpa att köra Project > Clean...

Faktorisering

Faktorisering innebär bl.a.

- Flytta/byta namn, m.m. på paket, klasser, metoder, variabler
- Ändra metodsignaturer
- Bryta ned stora metoder
- Ersätta klasser med interface
- ...

All faktorerering sköts från Eclipse (Markera > Högerklick > Refactor)

- Eclipse ändrar överallt i programmet så att allt blir korrekt
- Ändra aldrig direkt i filsystemet!

Eclipse Debug

För att avlusa väljer man ... Debug As > Java Application

- Eclipse byter då till debug perspektivet
- Se vidare workshop

Effektiv Eclipse

Låt Eclipse göra alla rutinjobb

- Importera
- Formatera (snygga till koden)
 - Ctrl + Shift + F
- Faktorisera
- Generera kod
 - Generera konstruktorer, set/get-metoder, m.m.
- Kommentera/avkommentera
- TODO lista

Om du behöver det så finns det i Eclipse, leta... (högerklicka i koden)

Eclipse Avslöjar Proffsen



Lär av proffsen

- Om man angivit var källkoden till Javas standardbibliotek finns (src.zip) kan man direkt granska klasser m.m.

Markera något klassnamn, tryck F3

- Otroligt nyttigt och intressant att se "professionell" kod
- Att se hur man implementerat saker ökar förståelsen

Eclipse Tutorials

Intro för nybörjare

<http://eclipsetutorial.sourceforge.net/totalbegginer01/lesson01.html>

Debug

<http://eclipsetutorial.sourceforge.net/debugger01/lesson01.html>

Eclipse Plugins

Hela Eclipse är uppbyggt av plugins. Olika versioner av Eclipse innebär egentligen olika uppsättningar av plugins.

I samma anda går det att installera extra plugins i den version du använder. De kan installeras på olika sätt, ett vanligt är via menyn

Help > Install New Software > Add

I dialogrutan anges adressen till en “Software (Update) Site” som plugins ska hämtas från. Denna behöver man känna till själv, t.ex. genom att kopiera adressen från en hemsida (se slides FindBugs och STAN). Av erfarenhet kan sägas att många “software site adresser” har en tendens att ändras så gå alltid in på tillverkarens hemsida för att vara säker på att få senaste versionen.

Välj därefter just de plugins som ska installeras, resten sköts automatisk, men du kan uppmanas att godkänna avtal och ev. installation av “unsigned software”.

Eclipse behöver normalt startas om efter installationen.

Plugins: FindBugs

Hemsida

<http://findbugs.sourceforge.net/>

Software site

<http://findbugs.cs.umd.edu/eclipse>

Körning

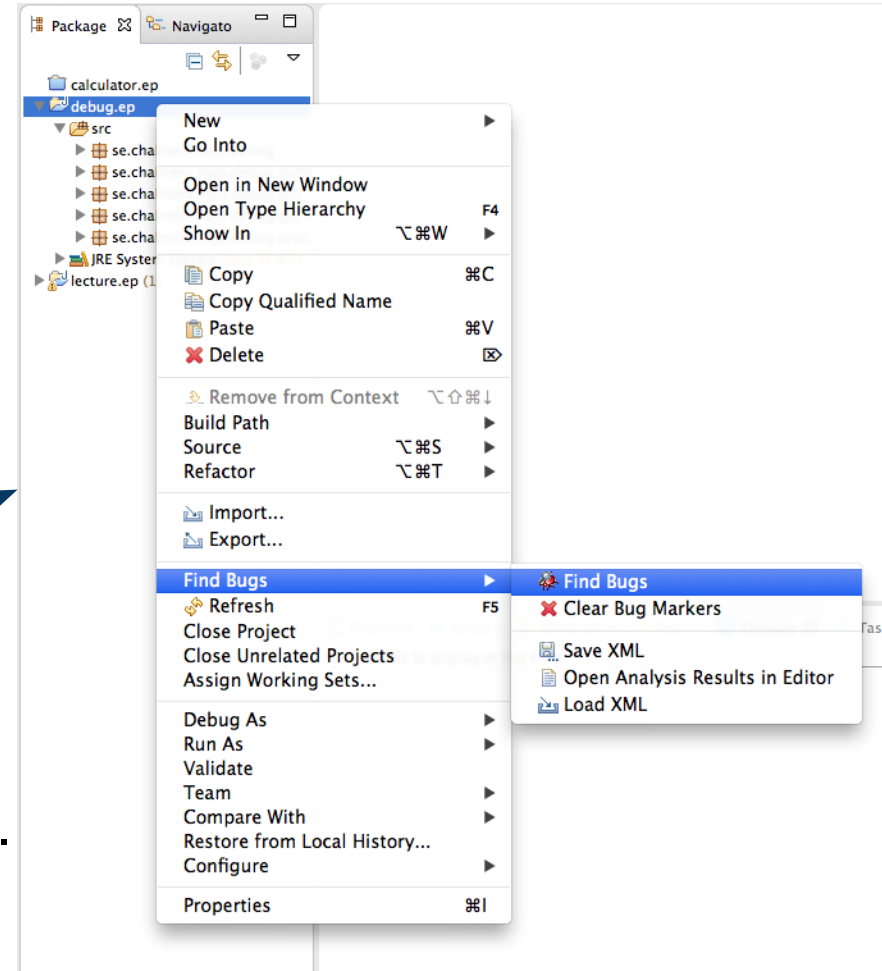
Högerklicka projektet och välj

Find Bugs > Find Bugs

Då öppnas ett eget perspektiv med flera egna vyer.

Syfte

Analyserar koden och listar både potentiella buggar och andra osäkerheter eller best practices som inte följts. Det går att välja vilka analyser som ska genomföras. Mycket användbart!



Plugins: STAN

Hemsida

<http://stan4j.com/>

Software site

<http://update.stan4j.com/ide>

(välj plugins “STAN IDE” samt “STAN IDE Tools”)

Körning

Högerklicka projektet och välj

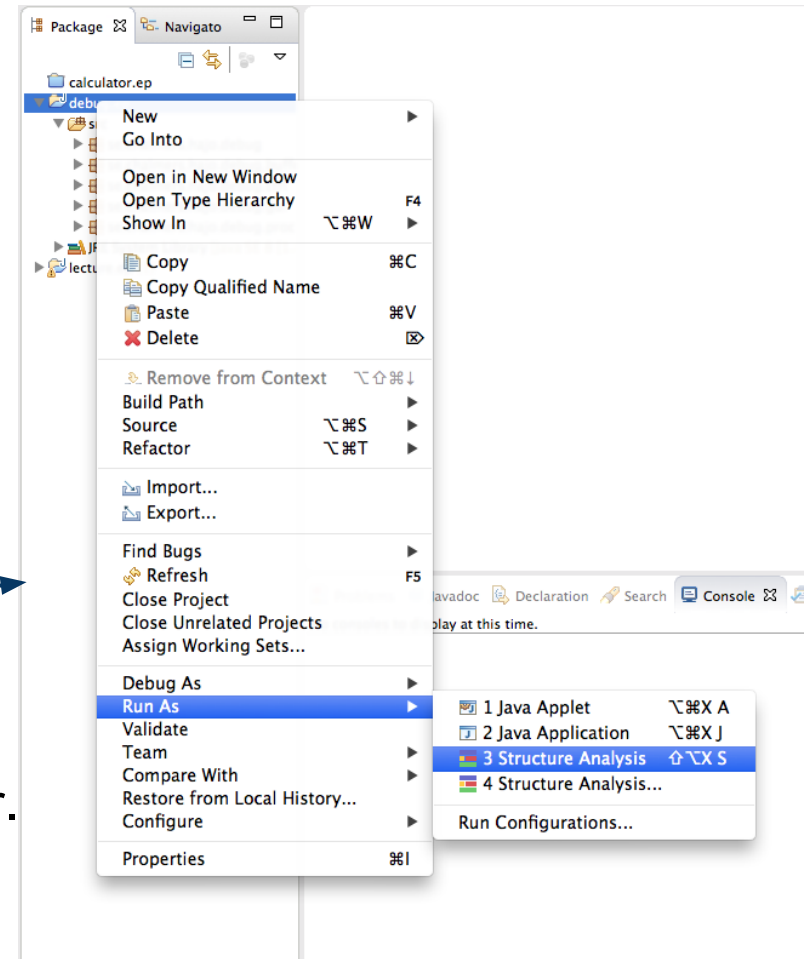
Run As > Structure Analysis

Då öppnas ett eget perspektiv med flera egna vyer.

Syfte

Används för strukturanalys:

- Beroenden mellan olika delar (paket, klasser etc) kan analyseras och koden kan utforskas interaktivt.
- Avancerad analys kan göras för att bedöma kodens kvalitet.
 - Detektion av cykliska beroenden
 - Mätvärden, t.ex. de framtagna av Robert C. Martin
 - För den intresserade finns ett white paper på: <http://stan4j.com/papers/stan-whitepaper.pdf>



Referens: Eclipse Keyboard Shortcuts

Här följer användbara tangentbordskommandon i Eclipse (bara som referens för den intresserade).

Alla shortcuts går att ställa in via menyn: Preferences->General->Keys

Finns även förinställda scheman ex. Emacs.

Notera att de kan skilja sig på Mac OS (oftast genom att Ctrl byts mot Command).

Debug/Körning

F5 - Step into

F6 - Step over

F7 - Step out

F11 - Starta senaste debug

Ctrl + F11 - Starta senaste körning

Navigera mellan vyer/editorer/perspektiv

Ctrl + F6 - Växla fil

Ctrl + E - Växla fil via filnamn

Ctrl + F7 - Växla vy

Ctrl + F8 - Växla perspektiv

Alt + Shift + Q, P - Visa/Växla till Package Explorer

Texteditor (avser ofta koden där markören står)

Håll musen över kod (hover) öppnar beskrivning/dokumentation

Ctrl + 1 - Quick fix menyn ("glödlampa")

Ctrl + 3 - Quick access menyn (testa!)

Ctrl + Mellanslag - Content assist / Code completion

Texteditor (avser ofta koden där markören står) forts.

Ctrl + f - Sök/Ersätt i aktuell fil

Ctrl + j - Inkrementell sökning i aktuell fil

Ctrl + h - Sök i hela Workspace/Projektet/etc.

Ctrl + l - Gå till rad

Ctrl + q - Gå till senaste editeringsposition

F3 - Öppna definitionen

F4 - Öppna typhierarkivyn

Ctrl + t - Öppna tillfällig typhierarki

Ctrl + - Vänster musklick Öppna definition (med val)

Ctrl + - Shift + G Lista referenser till kod

Ctrl + - Shift + T Öppna typ via namn

Ctrl + . - Hoppa till nästa kompileringsfel i aktuell fil

Ctrl + / - Kommentera/Avkommentera markerad kod

Ctrl + Shift + F - Korrigera indentering m.m. (formatera) hela filen

Ctrl + i - Korrigera indentering

Ctrl + Shift + O - Rensa upp i import

Se vidare

<http://www.vogella.com/tutorials/EclipseShortcuts/article.html>

<http://www.shortcutworld.com/en/win/Eclipse.html>