

Verktyg och Utvecklingsmiljö

Jochim von Hacht

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 och Findbugs

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...

- Chalmers version: Eclipse Helios(?)



Hämta hem egen version

<http://www.eclipse.org/downloads/> (välj Classic eller 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 katalogen som "Location" (lösa filer går inte)
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

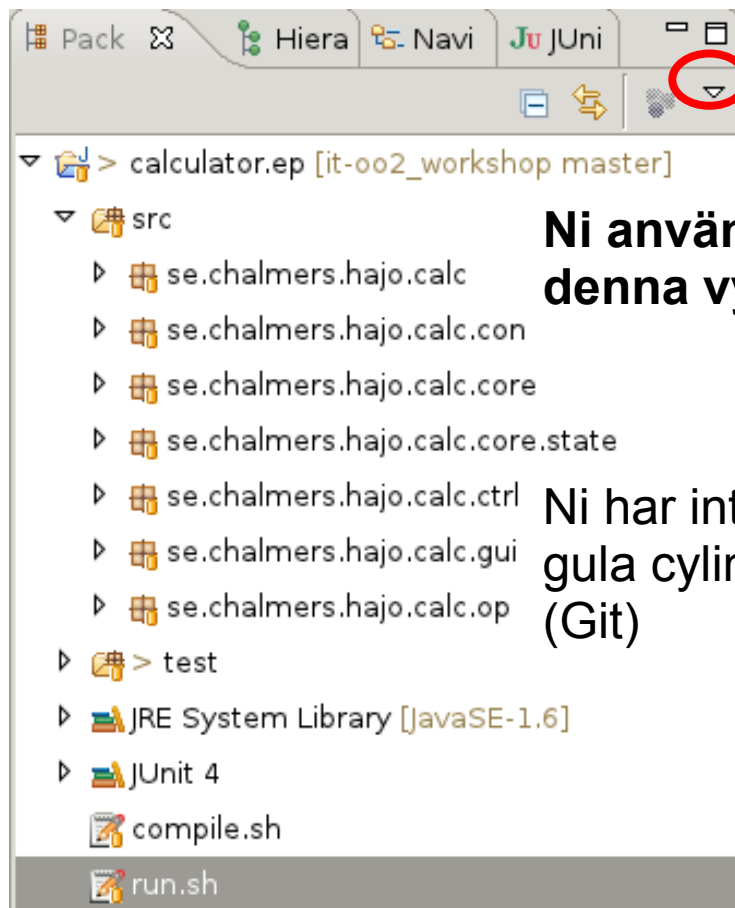
Perspective, ett visst arrangemang (layout) av ett visst antal views

- 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 väja under Window-menyn

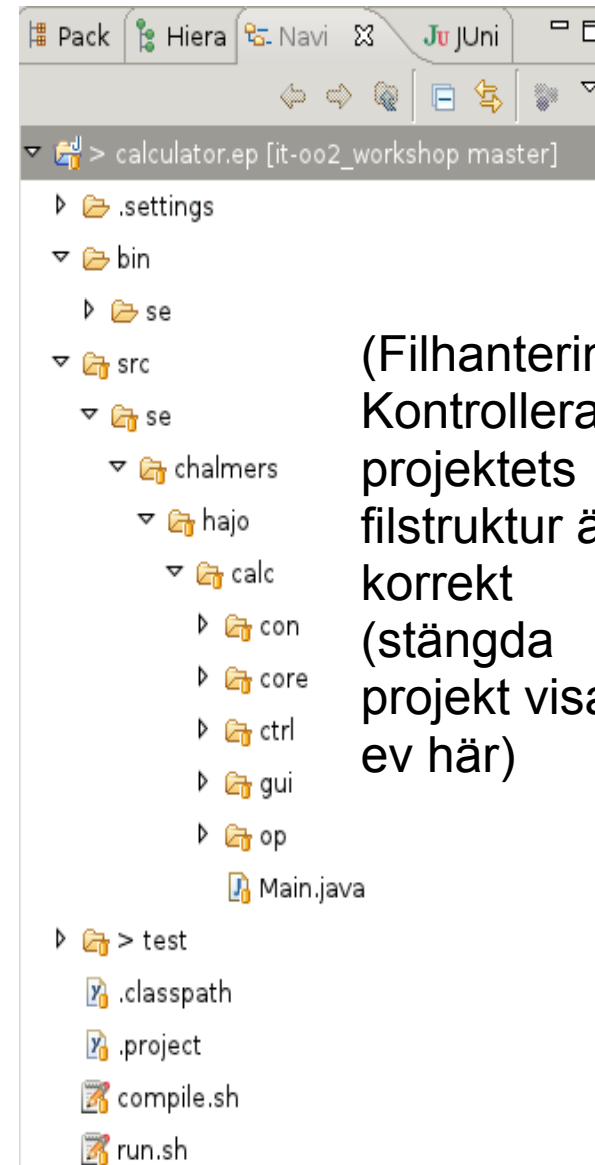
Olika Vyer av Projekt

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
- ...

Allt faktorerings 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)
- 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/totalbeginner01/lesson01.html>

Debug

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

STAN

Exempel på Eclipse-plugin <http://stan4j.com/eclipse/eclipse-integration.html>

För att installera plugins Help > Install New Software > Add (en länk till plugin:en) > ... resten sköts automatisk