

Utvecklingsverktyg (Eclipse)

Anton Ekberg

Planering

- Introduktion till utvecklingsverktyget Eclipse
- Övning
- Eclipse
 - Workspace
 - Koncept
 - Faktorisering
 - Avlusning
 - Tips
 - Plugins

Utvecklingsverktyg

Programutveckling består av ett antal sysslor vilka kräver verktyg

- Kodning
- Kompilering
- Testning
- Avlusning (debugging)
- Versionshantering
- Kodanalys

Integrated Development Environment, IDE

En gemensam plats för flera utvecklingsverktyg.

Istället för att ha ett specifikt verktyg för varje syssla.

Fördelar gentemot en vanlig textredigerare,

- Kodkompletering
- Syntax highlighting
- Felmeddelanden och varningar utan att själv kompilera sin kod

Exempel på felhantling i IDE vs javac

```
2  
3 public class Main {  
4     public static void main(String[] args) {  
5         throw new Exception();  
6     }  
7 }  
8
```

Main.java:5: error: unreported exception Exception; must be caught or declared to be thrown

```
    throw new Exception();  
    ^
```

1 error

Eclipse

En platform för IDE skriven i Java med stöd för många programmeringsspråk.

Utbyggbart med olika plugins

Förpacketerat i olika versioner för olika utvecklingsändamål

Går att ladda hem via www.eclipse.org/downloads/eclipse-packages och välj Eclipse IDE for Java Developers

Workspace

Ett workspace innehåller,

- Ett antal projekt
- Personliga inställningar för Eclipse

Möjligt att ha flera workspaces för att separera projekt.

Inte rekommenderat att spara sin källkod i workspace mappen.

Koncept i Eclipse

- ***Projekt***
Innehåller källkod och specifika inställningar.
- ***Editor***
Själva textredigeraren
- ***Perspective***
En samling av fönster (view) med ett syfte såsom att debugga
- ***View***
Fönsterelement som kan flyttas runt i huvudfönstret.

Faktorisering

Omstrukturering av kod

- Flytta/byta namn på paket, klasser, metoder, variabler mm.
- Bryta ned stora metoder
- Ändra metodsSignaturer

Eclipse har stöd för att utföra vissa refaktoriseringar automatiskt*

Högerklicka på en klass/metod och välj Refactor

Avlusning (debugging)

Eclipse har ett eget perspektiv för debugging

Högerklicka på ditt projekt och välj Debug as > Java Application

Sätt brytpunkter på intressanta rader/metoder genom att högerklicka på raden eller klicka i vänstermarginalen.

- F5 - Exekverar raden och går till nästa, om det är en metodanrop går den in.
- F6 - Exekverar raden/metodanropet utan att gå in metoden.
- F7 - Exekverar klart hela nuvarande metoden.
- F8 - Exekverar till nästa brytpunkt

Eclipse avslöjar proffsen

Eclipse har tillgång till källkoden för standardbiblioteket. (Om sökvägen till src.zip är angiven)

Genom att använda sig utav Open Declaration kan man läsa och lära sig av hur "professionella" programmerare strukturerar och skriver kod.

Fungerar för all kod som Eclipse känner till källkoden för (inte förkompilerad)

Kan även vara bra att känna till hitta alla referenser till en metod.

Plugins

Plugins bygger ut funktionaliteten hos Eclipse.

Installeras via Eclipse Marketplace eller manuellt.

Help > Eclipse Marketplace / Install new software.

För manuell installation lägger man till en ny “site” som hittas på utvecklarens hemsida och därefter väljer den version man vill installera.

Eclipse kan behövas startas om efter installationen.

FindBugs och STAN

- FindBugs

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

Hjälper till att hitta buggar och att följa best practices.

- STAN - STatic ANalysis

<http://stan4j.com/>

Utför statisk kodanalys för att hitta beroenden och bedöma kodkvalitet

Användbara länkar

- Introduktion till Eclipse

<https://www.youtube.com/playlist?list=PLv6UtFrA7VEu4PtzJaGHHSeZBi6mdJtwv>

- Eclipse debugging

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

<https://www.youtube.com/playlist?list=PLUGinutrSzN-ALgaL0mSyPDHi3VhUCcuV>

- Kortkommandon i Eclipse

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