



TDP005, Projekt: objektorienterade system

Laboration: Eclipse

Höstterminen 2011

Revision 2



Introduktion

Detta material innehåller övningar för Eclipse.

Redovisning

Dessa övningar behöver **inte** redovisas, men ni får gärna stämma av med en assistent när ni är klara.

Utvecklingsmiljö

I kursen används Eclipse som utvecklingsmiljö och Subversion som revisionshanteringssystem. Subversion bör du ha installerat sedan tidigare. Nedan är instruktioner för att installera Eclipse.

Installation

Java

Som standard används libgcj som javamotor (java runtime environment) i Ubuntu. Tyvärr kan det ställa till det för Eclipse och därför vill vi installera Suns JRE (eller alternativt OpenJDK, om man hellre vill det).

Installera paketet 'sun-java6-jre'. För att kontrollera vilken java-version som används kan du köra:

```
$ java -version
```

För att se tillgängliga kör du:

```
$ sudo update-java-alternatives --list
```

För att byta kör du:

```
$ sudo update-java-alternatives -s {JVMNAMN}
```

där du byter ut {JVMNAMN} mot till exempel java-6-sun.

Eclipse

Installera Eclipse med Ubuntus pakethanteringssystem (till exempel med *Synaptic*), följ steg 1a. För att få en senare version så kan man installera manuellt. Följ då steg 1b.

Steg 1a

Synaptic installerar för närvarande version 3.5.2 av Eclipse. Detta steg görs mest för att få in de paket Eclipse beror på. Du kan låta bli om du vill men får då själv se till så att behövda paket installeras (se vidare nedan (*)).

Installera CDT. CDT är en C++-plugin till Eclipse. Den installeras inuti Eclipse:

- Välj *Help --> Install new software*
- Välj *Galileo Update Site*
- Under *Programming Languages* så hittar ni *Eclipse C/C++ development tools*.
- Installera, och starta om Eclipse.

Steg 1b

Installera Eclipse manuellt (för att få en senare version, version 3.7):

- ladda ner en tarboll från Eclipse hemsida:
www.eclipse.org --> Downloads --> Eclipse for C++, Linux 32bit
- packa upp filen och lägg förslagsvis mappen 'eclipse' i '~/opt/'
(finns inte '~/opt/' så skapa den)
- öppna mappen '~/opt/eclipse' och dra filen 'eclipse' till ditt verktygsfält högst upp på skärmen
(gnome toolbar) så kan du enkelt starta Eclipse därifrån

(*) För att kunna kompilera C++-program behöver du en kompilator för ändamålet. Installerar du endast Eclipse manuellt enligt ovan så måste du själv se till så att du har en kompilator. Installera paketet 'g++' så är det fixat! Installera 'gdb' för att kunna debugga program. Se även till så att du har paketet 'make' installerat samt 'exuberant-ctags' (för indexering).

Steg 2

Starta Eclipse (välj en katalog för workspace – inte viktigt just nu, det går lätt att byta senare) och gå till *Workbench*. Uppdatera:

- Välj *Help* --> *Check for updates*
- Välj *Update...*
(vänta på nedladdning)
- Starta uppdateringen (Först *Next*, sen acceptera avtal, sen *Finish*)
- Starta om Eclipse

Uppdatering klar!

Subclipse

Subclipse är en subversion-plugin för Eclipse. För att installera, gör följande (instruktioner för Eclipse version 3.7):

1. Välj *Help* --> *Install new software*
2. Välj fliken "Available Software Sites" och lägg sedan till följande under "Add":
http://subclipse.tigris.org/update_1.6.x
3. Nu kan du markera följande för installation:
 - Subclipse
 - Core SVNKit Library
4. Installera!
Det kan ta ett tag – kika i statusfältet i Eclipse huvudfönster om du undrar vad som händer.

Du behöver också installera JavaHL. Det gör du genom att installera paketet *libsvn-java* med apt-get.

Nu har du installerat Subclipse och kan börja använda subversion direkt i Eclipse! Först bör du (om du inte redan har gjort det) skapa en katalogstruktur för projektet i ditt svn-repo.

Användning

För att börja använda Subclipse i ett nytt projekt så gör följande:

1. Välj *File --> New --> Other* (eller tryck shift-alt-N)
2. *SVN --> Checkout Projects from SVN*
3. Lägg till URL för ditt repository (rot-katalogen)
4. Välj katalog från repositoryt att använda för projektet (högerklicka för att kunna uppdatera vyn)
5. Använd ”Check out as project configured using the New Project Wizard” (förvalt)
6. Skapa ett C++-projekt:
C++ --> Executable --> Empty Project

Nu kan du högerklicka på ett projekt och välja ”Team” för att hitta subversion-kommandon!

CUTE

Om du vill kan du installera CUTE. För att göra detta, lägg till följande site:

- <http://www.cute-test.com/updatesite>

på samma sätt som för Subclipse. Markera och installera! Se kurshemsidorna för mer info om CUTE (TDP004 och TDP005).

Övning – Eclipse

Syftet med övningen är att ge en introduktion till Eclipse och CDT. Syftet är också att ni ska känna er bekväma med att börja använda Eclipse som editor samt att ni har tillräckligt med förståelse om Eclipse för att kunna utöka era kunskaper om verktyget på egen hand.

1. Öppna Eclipse och välj en Workspace (platsen där alla era C++-projekt ni skriver i Eclipse hamnar)
2. Första gången ni öppnar Eclipse hamnar ni i ett välkomstperspektiv. För att komma vidare till "standardperspektivet" gå vidare via "go to the workbench" (motsvarande).
3. Skapa ett nytt C++-projekt. Detta kan du göra på flera sätt, till exempel genom att använda huvudmenyn: *File --> New --> Project*.
Välj sedan *C++ --> Executable --> Empty Project*. Du får då upp en dialogruta där du får döpa projektet. Döp det till `test1`. Övriga inställningar behöver inte ändras.
4. Skapa en ny C++-fil genom att högerklicka på ditt projekt och välja *New --> Source File*. Döp filen till `hello.cpp`.
Skriv in koden för ett main-program som skriver ut texten "Hello world!" i filen.
5. Om något är fel i koden visar Eclipse det genom en liten röd ikon i vänsterkanten. Håll musen över ikonen för att se vad felet är. Om du inte fått några fel, prova att införa några, till exempel genom att ta bort semikolon, stava fel, inte stänga paranteser eller "glömma" inkludera paket.
Man kan även få varningar, en liten gul ikon. Prova till exempel att deklarera en variabel som ni sedan inte använder.
6. Kör programmet.
Det gör du till exempel genom att välja *Run --> Run (Ctrl-F11)* i huvudmenyn.
7. Om du vill köra ditt program igen kan du göra det genom att klicka på den lilla gröna ikonen med en pil, under huvudmenyn.
8. Det mesta i Eclipse går att göra på flera olika sätt. Prova att skapa ett nytt projekt med ett enkelt program, men utför kommandona på något annat sätt. Använd till exempel huvudmenyn om du inte gjort det innan, eller den kontextkänsliga menyn som dyker upp när man högerklickar på olika ställen i Eclipse.
9. Eclipse kan ge dig stöd på olika sätt när du programmerar. Några av dem ska vi titta på nu.
 - Deklarera en sträng i ditt program och ge den ditt namn som värde:

```
string namn;  
namn = "Sara";
```

Ändra i koden så att "Hello [ditt namn]" skrivs ut istället för "Hello world!" som tidigare. Högerklicka på variabeln `namn`. Välj *Refactor* i menyn och byt namn på variabeln till `name`.
Notera att variabelns namn byts ut på alla ställen i koden.
 - Skriv i filen (inklusive punkten):

```
namn.
```

Då får du upp en ruta med förslag på vad som kan göras med en sträng. Titta gärna igenom alla förslag. Välj sedan `clear`. Skriv ut strängen igen. Vad gjorde `clear`?

- Skriv in bokstäverna `get` i filen. Högerklicka sedan på dem och välj *Source* --> *Content Assist* i menyn (Ctrl-Space). Då visas en lista med alla tillgängliga satser som börjar med "get". Titta igenom den, och välj sedan den första förekomsten av `getline`. Lägg till parametrar:

```
getline(cin, name);
```

Skriv ut namn igen. Vad händer när du kör programmet? Jo, det väntar på att du ska skriva in en text. Gör det! Och se vad utskriften blir.
 - Prova att använda *Content Assist* på andra bokstavskombinationer och varianter. Skriv till exempel `cout << e` och tryck Ctrl-Space.
10. Det är fortfarande tidigt i kursen och objektorientering har inte kommit upp, men bara för att få en känsla för vad Eclipse kan åstadkomma ska vi prova att skapa en klass. Högerklicka på projektnamnet och välj *New* --> *Class*. Ge klassen ett godtyckligt namn. Titta på filerna som Eclipse skapar för att se vad Eclipse har genererat.
 11. Ctrl-Shift-F formaterar enligt vald kodstil (vilket i kursen ska vara K&R – den förvalda). Testa att lägga in radbrytningar lite här och var mellan identifierare och måsvingar och testa sedan kortkommandot (Det blir kanske inte helt perfekt men förhoppningsvis bättre).
 12. Ctrl-. (punkt) och Ctrl-, (komma) förflyttar framåt och bakåt mellan bland annat kompilersfel i en kodfil (röda understrykningar). Kolla i *Edit* (huvudmenyn) för fler kortkommandon och även i kontextmenyn (högerklicka) under *Source*.
 13. Tryck F11. Nu får du frågan om du vill byta till ett nytt perspektiv, nämligen perspektivet för debuggning. Svara ja och se hur Eclipse formar om sig till din stora ära! Öhm, jag menar: lägg märke till perspektiv-fliken och perspektiv-knappen uppe i högra hörnet av Eclipse. Där kan du byta mellan olika perspektiv.
Om du vill kan du nu utforska debugg-perspektivet genom att till exempel stega i programmet med F5 (*step into*) eller F6 (*step over*). Fler möjligheter finns – kolla i fliken Debug uppe till vänster i Eclipse.
Tryck F8 för att återgå till att köra programmet som vanligt. Byt sedan tillbaka till vanligt perspektiv.
 14. Detta var bara ett smakprov på vad som går att göra med Eclipse. Prova gärna att göra fler saker, undersök vilka val som finns i de kontextkänsliga menyerna man får upp när man högerklickar etcetera.
 15. När du känner dig klar, visa upp dina projekt för labbassistenten, och var beredd att svara på frågor kring det du gjort i övningen.

Att köra ett enkelt program i Eclipse med SVN

Starta Eclipse. Skriv programmet nedan och spara det på en fil med namnet `uppg1.cpp`. Har du följt eclipse-instruktionerna så har du redan en katalogstruktur för dina labbar och övningar annars bör du fixa det nu:

Följande kommando skapar en katalog i din SVN repository:

```
$ svn mkdir --parents URL/eclipse-ovn
```

Där URL är pathen till din SVN-area

Finns inte `eclipse-ovn` så skapas den (`svn help mkdir;-)`)

Skapa projektet:

- *File --> New --> Other...*
- *SVN --> Checkout Projects from SVN*
- Välj repository (lägg till om nödvändigt)
- Välj katalogen `eclipse-ovn`
- Gå vidare med det förvalda ”*Check out as a project configured using the New Project Wizard*”
- Välj *C++ --> C++ Project*
- Skriv ett projektnamn, till exempel `uppg1`, och välj sedan *Executable --> Empty Project*
- Gå vidare med *Finish*

Du har nu ett tomt projekt som du kan skapa nya filer i. Skapa filen `uppg1.cpp` genom att:

- Välja *File --> New --> Source Folder* och skapa katalogen `src`. Detta steg är inte nödvändigt, speciellt inte i introduktionsövningarna (vi skapar ju bara en kodfile per projekt), men trevligt annars då vi kommer ha flera kodfiler. Du kan ju testa en gång för att se hur det funkar men skippa det för övriga ”programX”-projekt.
- Välja *File --> New --> Source File* skapa filen `uppg1.cpp`. Om du har en `src`-katalog se till så att filen hamnar däri.

uppg1.cpp

```
/*
 * Program som skriver ett meddelande på standard utmatningsfil.
 */
#include <iostream>
using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    return 0;
}
```

Eclipse

För att kompilera och länka kan du till exempel välja *Project --> Build Project*. För att köra kan du trycka på den gröna ”play”-knappen i verktygsfältet eller välja *Run --> Run* (Ctrl+F11).

Subclipse

Högerklicka på projektet i *Project Explorer* och välj *Team --> Commit* och gå vidare där för att lägga in projektet i subversionrepositoryt. Skriv en kommentar och kolla vilka filer som ändrats i och som kommer läggas till.

I *Project Explorer* kan du nu se revisionsnummer, datum, klockslag och vem som ändrat i filen till höger om namnet (inom hakparenteser och i en lite gulare färg). För projektet visas vilken katalog den ligger i relativt roten för repositoryt.

Eclipse

För att städa i projektet välj *Project --> Cleanup...* När man städar ett projekt tas de filer bort som genereras när man bygger projektet (till exempel *.o).