

Laboration 1

Avsikt

Att bekanta dig med NetBeans programmeringsmiljö och skriva några enkla program med programmeringsspråket Java.

Del 1

På kursen ska du använda *Eclipse* för att utveckla java-program i. Ett alternativ till *Eclipse* är *NetBeans* som vi använt på tidigare kurser. Ta fram dokumentet **Eclipse.pdf** (eller **NetBeans6_1.pdf**) från kurssidan och arbeta med dokumentet. När du är färdig fortsätter du med Del 2.

Del 2

I denna del ska skriva ett nytt program och sedan ändra i det. Men först några mer utförliga kommentarer runt det sista programmet i Del 1.

```
public class EttProgram {  
    public void action() {  
        String name = JOptionPane.showInputDialog( "Ange ditt namn" );  
        System.out.println("-----");  
        System.out.println("Ett program skrivet av " + name );  
        System.out.println("-----");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        EttProgram prog = new EttProgram();  
        prog.action();  
        javax.swing.JOptionPane.showMessageDialog( null, "Nu är programmet slut!" );  
    }  
}
```

Följande sker vid programkörningen

- 1) Programkörningen startar i **main**-metoden med programraden

```
EttProgram prog = new EttProgram();
```

`new EttProgram()` skapar ett objekt av typen **EttProgram**. Det innebär bl.a. att de metoder som finns i klassen **EttProgram** kan anropas. *Referensvariabeln* **prog** håller reda på det skapade objektet. Anrop av metoden `action` i **EttProgram** sker med hjälp av **prog** (se punkt 2).

- 2) Därefter utförs programraden

```
prog.action();
```

Denna rad innehåller ett anrop till metoden **action** (sker med referensvariabeln **prog** följt av en punkt och sedan metodens namn). Anropet innebär att programexekveringen hoppar till metoden **action** och programraderna i **action** kommer att utföras.

3) Programraderna

```
String name = JOptionPane.showInputDialog( "Ange ditt namn" );  
System.out.println( "-----" );  
System.out.println( "Ett program skrivet av " + name );  
System.out.println( "-----" );
```

utförs i sekvens uppifrån och ner.

Raden

```
String name = JOptionPane.showInputDialog( "Ange ditt namn" );
```

innehåller flera delar.

Högerledet innebär ett anrop till metoden `showInputDialog` som finns placerad i klassen `JOptionPane`. Vid anropet visas en dialog i vilken användaren kan mata in text. Den text användaren matar in blir resultatet av anropet och detta resultat lagras i variabeln `name`.

Raden

```
System.out.println("-----");
```

skriver ut en rad med korta streck.

Raden

```
System.out.println("Ett program skrivet av "+name);
```

skriver ut texten "Ett program skrivet av " följt av innehållet i variabeln `name`.

- 4) Nu finns det inte fler programrader att utföra i **action**-metoden så nu återgår programexekveringen till **main**-metoden. Nästa programrad att utföra är programraden efter anropet till **action**. Här anropas metoden `showMessageDialog` som finns i klassen `JOptionPane`. Vid anropet till `showMessageDialog` visar sig en dialog med texten "Nu är programmet slut!".

Nytt program

När du ska skriva ett helt nytt program så börjar du med att skapa ett nytt projekt. Sedan skapar du successivt de klasser som ska ingå i programmet.

Varje program du skriver kommer att bestå av minst två klasser, nämligen en klass som innehåller koden (EttProgram ovan) och en klass med vars hjälp exekveringen startar (Main ovan). Detta gäller i vart fall under första delen av kursen. Du kan ge klasserna vilka namn som helst men:

- 1 De ska alltid börja med stor bokstav
- 2 De ska ha meningsfulla namn
- 3 De ska endast innehålla bokstäver och siffror

Program 2

Nu ska du skriva ett program som

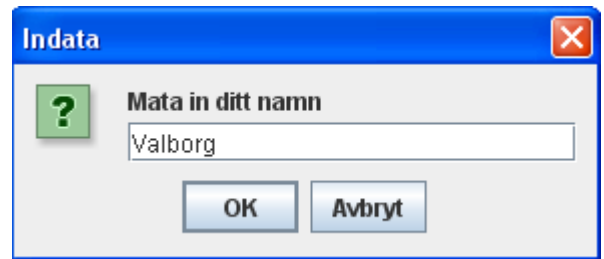
1. Talar om vad programmet heter.
2. Låter en användare mata in sitt namn och sedan skriver ut namnet med stora bokstäver.
3. Tackar för att användaren använt programmet.

Så här kan en programkörning se ut:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Valborg med stora bokstäver: VALBORG

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X  Tack för att du använde programmet  X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```



Gör så här:

- Skapa projektet **laboration1** i katalogen som du valt som projektkatalog.
- Skapa klassen **Uppercase**. Uppercase ska tillhöra projektet laboration1. Skriv in koden i nedanstående listning.

```
import javax.swing.JOptionPane;

public class Uppercase {
    public void uppercaseEx() {
        String name;
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
        System.out.println( "X                VERSALER                      X" );
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
        System.out.println();
        name = JOptionPane.showInputDialog( "Mata in ditt namn" );
        System.out.println( name + " med stora bokstäver: " + name.toUpperCase() );
        System.out.println();
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
        System.out.println( "X  Tack för att du använde programmet  X" );
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
    }
}
```

- Skapa klassen **StartUppercase** vilken ska innehålla en main-metod vilken anropar metoden uppercaseEx. Skriv in koden i nedanstående listning.

```
/*
 * StartUppercase.java
 * Skapad den 30 augusti 2008, 22:26
 */

/**
 * Programmet ger exempel på ovandling till stora bokstäver
 * @author Rolf Axelsson
 */
public class StartUppercase {
    public static void main( String[] args ) {
        Uppercase prog = new Uppercase();
        prog.uppercaseEx();
    }
}
```

- Kör programmet och studera körresultatet. Stämmer körresultatet med programraderna som utförs?

Ändra i programmet

Det går utmärkt att skriva flera metoder i samma klass. Ändra i klassen **Uppercase** så att den ser ut så här:

```
import javax.swing.JOptionPane;

public class Uppercase {
    public void presentation() {
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
        System.out.println( "X                VERSALER                X" );
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
    }

    public void versaler() {
        String name = JOptionPane.showInputDialog( "Mata in ditt namn" );
        System.out.println();
        System.out.println( name + " med stora bokstäver: " + name.toUpperCase() );
        System.out.println();
    }

    public void avslutning() {
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
        System.out.println( "X    Tack för att du använde programmet    X" );
        System.out.println( "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX" );
    }
}
```

Nu går det inte att anropa metoden **uppercaseEx** i klassen **Uppercase**. Den innehåller ju inte någon sådan metod. Men metoderna **presentation**, **versaler** och **avslutning** går utmärkt att anropa.

Ta bort anropet

```
prog.uppercaseEx();
```

ur main-metoden och skriv in nedanstående anrop.

```
prog.presentation();
prog.avslutning();
prog.versaler();
```

Kör programmet och fundera över körresultatet. Ändra sedan ordningen på anropen så att körresultatet blir samma som det var tidigare (innan vi gjorde ändringarna).

Anropa metoder i den egna klassen

Nu ska du på nytt lägga till metoden **uppercaseEx()** i klassen **Uppercase**. Tanken är att ett anrop till **uppercaseEx** ska ge samma körresultat som tidigare. För att åstadkomma detta ska metoden se ut så här.

```
public void uppercaseEx() {
    presentation();
    versaler();
    avslutning();
}
```

Kontrollera att körresultatet är korrekt genom att anropa **uppercaseEx** från main-metoden:

```
prog.uppercaseEx();
```

Som du ser anropar man enkelt metoder i samma klass.

Ändra i programmet

Ändra i main-metoden så du får nedanstående körresultat. Du ska använda dig av metoderna presentation, versaler och avslutning. Det går naturligtvis bra att anropa samma metod flera gånger.

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                      VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Gunnar med stora bokstäver: GUNNAR

Valborg med stora bokstäver: VALBORG

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X  Tack för att du använde programmet  X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Ändra i programmet

Ändra i main-metoden så du får följande körresultat:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X  Tack för att du använde programmet  X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Mata in ditt namn: Kristian

Kristian med stora bokstäver: KRISTIAN

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                      VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Ändra i programmet

Ändra i main-metoden så du får följande körresultat:

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                      VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X  Tack för att du använde programmet  X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                      VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X                      VERSALER                      X
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

