

Kapittel 15: Grafiske brukergrensesnitt: Enkel GUI

Del I

Redigert av:

Khalid Azim Mughal (khalid@ii.uib.no)

Kilde:

Java som første programmeringsspråk (3. utgave)

Khalid Azim Mughal, Torill Hamre, Rolf W. Rasmussen
Cappelen Akademisk Forlag, 2006.

ISBN: 82-02-24554-0

<http://www.ii.uib.no/~khalid/jfps3u/>

(NB! Boken dekker opptil Java 6, men notatene er oppdatert til Java 7.)

Emneoversikt

- Enkel dialogutforming med `javax.swing.JOptionPane`
 - Presentasjon av utdata til brukeren
 - Lesing av inndata fra brukeren
 - Bekreftelse av opplysninger fra brukeren
-

Enkel dialogutforming med JOptionPane

- Klassen JOptionPane tilbyr predefinerte dialogvinduer som kan brukes til å utveksle inndata og utdata med brukeren via et enkelt grafisk brukergrensesnitt.
- Utforming av dialogvinduer for tre formål ved hjelp av klassen JOptionPane:
 - Hvordan informasjon kan presenteres til brukeren
 - Hvordan brukeren kan taste inndata til programmet
 - Hvordan programmet kan be brukeren om å bekrefte opplysninger
- Klassen JOptionPane definerer tre statiske metoder som har navnet `showTypeDialog()`, der *Type* kan erstattes med `Message`, `Input` eller `Confirm`, avhengig av hvilken type dialogvindu vi ønsker.
- Alle dialogvinduene blir modale, det vil si at brukeren må bli ferdig med dialogvinduet før hun kan fortsette med programmet.
- GUI-basert programmet må avsluttes med kallet `System.exit(0)` i kildekoden.

Tabell 15.1: Utdrag av metoder fra klassen JOptionPane

javax.swing.JOptionPane	
<pre>static void showMessageDialog(Component foreldreKomponent, Object beskjed)</pre>	<p>Metoden showMessageDialog() brukes til å presentere informasjon til brukeren. Den første metoden bruker vindustittelen "Message" og beskjedtypen angitt ved JOptionPane.INFORMATION_MESSAGE.</p>
<pre>static void showMessageDialog(Component foreldreKomponent, Object beskjed, String tittel, int beskjedType)</pre>	
<pre>static String showInputDialog(Object beskjed)</pre>	<p>Metoden showInputDialog() brukes til å be brukeren om inndata. Det som ble tastet blir returnert som en streng. Dersom dialogvinduet ble kansellert, returneres null-litteralen. Den første og andre metoden bruker vindustittelen "Input" og beskjedtypen angitt ved JOptionPane.QUESTION_MESSAGE.</p>
<pre>static String showInputDialog(Component foreldreKomponent, Object beskjed)</pre>	
<pre>static String showInputDialog(Component foreldreKomponent, Object beskjed, String tittel, int beskjedType)</pre>	

javax.swing.JOptionPane

```
static int showConfirmDialog(  
    Component foreldreKomponent,  
    Object beskjed)
```

```
static int showConfirmDialog(  
    Component foreldreKomponent,  
    Object beskjed,  
    String tittel,  
    int opsjonsType)
```

```
static int showConfirmDialog(  
    Component foreldreKomponent,  
    Object beskjed,  
    String tittel,  
    int opsjonsType,  
    int beskjedType)
```

Metoden `showConfirmDialog()` brukes til å be brukeren om å bekrefte informasjon. For tolking av returverdien, se Tabell 15.4.

Den første metoden bruker vindustittelen "Select an Option", opsjonstypen angitt ved `JOptionPane.YES_NO_CANCEL_OPTION` og beskjedtypen angitt ved `JOptionPane.QUESTION_MESSAGE`.

Den andre metoden bruker beskjedtypen angitt ved `JOptionPane.QUESTION_MESSAGE`.

Tabell 15.2: Felles parametere til metodene `showTypeDialog()`

Parameternavn	Beskrivelse
Component foreldreKomponent	Angir rammen dialogvinduet skal plasseres i. En standard ramme opprettes dersom verdien er null eller ikke er spesifisert.
Object beskjed	Angir det som skal presenteres til brukeren, for eksempel en beskjed eller ledetekst. Vanligvis er dette et String-objekt. Dersom dette ikke er tilfellet, brukes <code>toString()</code> -metoden på objektet for å lage en stren-grepresentasjon som vises i dialogvinduet.
String tittel	Angir tittel som skal settes på dialogvinduet.
int beskjedType	Denne verdien angir implisitt hvilket ikon som skal brukes i dialogvinduet. Oversikt over lovlige verdier er gitt i Tabell 15.3.

Tabell 15.3: Angivelse av ikon i klassen JOptionPane

Beskjedtyper i javax.swing.JOptionPane	
ERROR_MESSAGE INFORMATION_MESSAGE WARNING_MESSAGE QUESTION_MESSAGE PLAIN_MESSAGE	Disse konstantene i klassen JOptionPane representerer standardikoner som brukes i et dialogvindu.

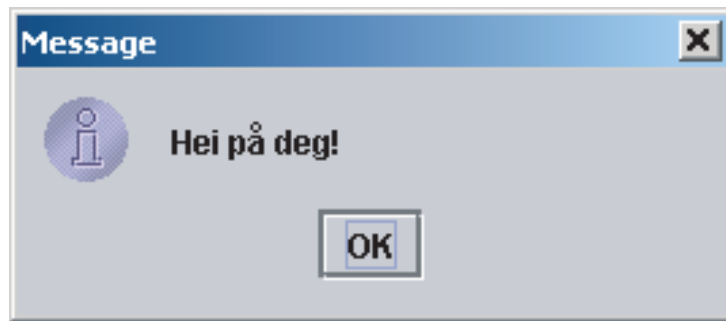
Tabell 15.4: Tolking av heltallsverdien returnert fra metodene i klassen JOptionPane

Returverdikonstant er definert i klassen JOptionPane	Angir hvilken handling brukeren har foretatt.
YES_OPTION	Klikket på Yes-knappen.
NO_OPTION	Klikket på No-knappen.
CANCEL_OPTION	Klikket på Cancel-knappen.
OK_OPTION	Klikket på Ok-knappen.
CLOSED_OPTION	Klikket på lukk-vindu-boksen til dialogvinduet.

Presentasjon av utdata til brukeren

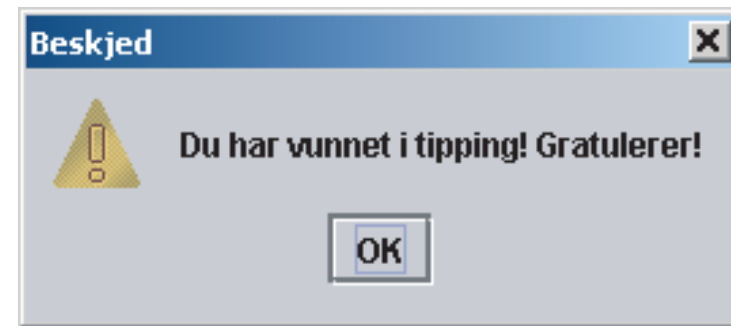
- Et slikt dialogvindu består vanligvis av en beskjed til brukeren og en OK-knapp som brukeren kan klikke på etter å ha lest beskjeden.
- Metoden `showMessageDialog()` brukes for slike dialogvinduer (Program 15.1).

Figur 15.1: Dialogvinduer med metoden `showMessageDialog()`



```
JOptionPane.showMessageDialog( // (1)  
    null, "Hei på deg!");
```

(a)



```
JOptionPane.showMessageDialog(null, // (2)  
    "Du har vunnet i tipping! Gratulerer!",  
    "Beskjed", JOptionPane.WARNING_MESSAGE);
```

(b)

Program 15.1: Bruk av metoden showMessageDialog()

```
import javax.swing.JOptionPane; // Importerer klassen JOptionPane

public class BeskjedDialog {
    public static void main( String[] args ) {

        JOptionPane.showMessageDialog(           // (1)
            null,                                // Ingen forelder-ramme
            "Hei på deg!"                        // Beskjeden
        );

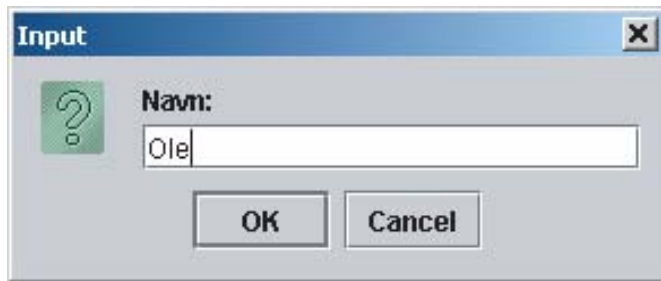
        JOptionPane.showMessageDialog(           // (2)
            null,                                // Ingen foreldre-ramme
            "Du har vunnet i tipping! Gratulerer!", // Beskjeden
            "Beskjed",                           // Tittel på vinduet
            JOptionPane.WARNING_MESSAGE          // Beskjedtypen
        );

        System.exit(0); // (3) Nødvendig for å avslutte programmet.
    }
}
```

Lesing av inndata fra brukeren

- Denne typen dialogvindu består vanligvis av et tekstfelt som brukeren kan skrive i og to knapper (en Ok-knapp og en Cancel-knapp) for å avlevere inndata eller kansellere dialogvinduet.
- Metoden `showInputDialog()` brukes for dette formålet (Program 15.2).
 - Metoden returnerer innholdet av tekstfeltet som en streng, som må eksplisitt konverteres til andre typer verdier, dersom programmet krever det.

Figur 15.2: Dialogvinduer med metoden `showInputDialog()`



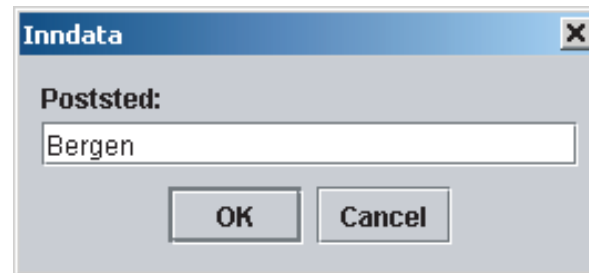
```
JOptionPane.showInputDialog("Navn:"); // (1)
```

(a)



```
JOptionPane.showInputDialog( // (2)  
    null, "Postkode:");
```

(b)



```
JOptionPane.showInputDialog( // (4)  
    null, "Poststed:",  
    "Inndata", JOptionPane.PLAIN_MESSAGE);
```

(c)

Program 15.2: Bruk av metoden `showInputDialog()`

```
import javax.swing.JOptionPane;

public class InndataDialog {
    public static void main( String[] args ) {
        String navn = JOptionPane.showInputDialog(           // (1)
            "Navn:"                                           // Ledeteksten
        );
        String postkodeStr = JOptionPane.showInputDialog(    // (2)
            null,                                             // Ingen foreldre-ramme
            "Postkode:"                                       // Ledeteksten
        );
        int postkode = Integer.parseInt(postkodeStr);        // (3)

        String poststed = JOptionPane.showInputDialog(       // (4)
            null,                                             // Ingen foreldre-ramme
            "Poststed:",                                       // Ledeteksten
            "Inndata",                                         // Tittel på vinduet
            JOptionPane.PLAIN_MESSAGE                         // Beskjedtypen
        );
        JOptionPane.showMessageDialog( null,
            navn + "\n" + postkode + " " + poststed,
            "Opplysninger", JOptionPane.PLAIN_MESSAGE);
        System.exit(0);
    }
}
```

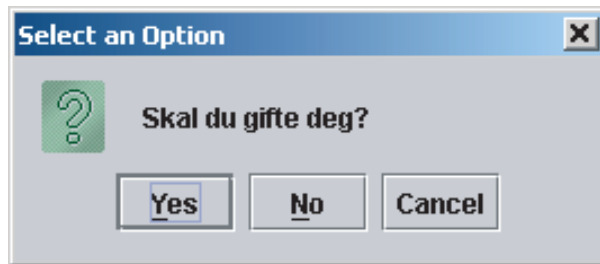
Bekreftelse av opplysninger fra brukeren

- Et slikt dialogvindu består vanligvis av et spørsmål om et forhold som brukeren må ta stilling til.
 - I tillegg har dialogvinduet vanligvis to knapper (en Yes-knapp og en No-knapp) for å gi svar på spørsmålet.
- Metoden `showConfirmDialog()` brukes for dette formålet (Program 15.3).
 - Tolking av returverdien fra metoden `showConfirmDialog()` er vist i Tabell 15.4.
 - Denne metoden kan også ta en parameter som angir *opsjonstype* (Tabell 15.5).

Tabell 15.5: Angivelse av knapper i metoden `showConfirmDialog()`

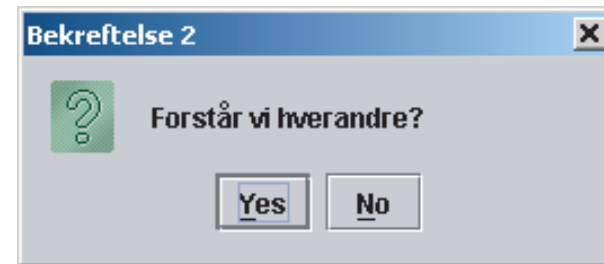
Opsjonstype definert i klassen <code>javax.swing.JOptionPane</code>	
<code>DEFAULT_OPTION</code>	Ok-knapp
<code>YES_NO_OPTION</code>	Yes- og No-knapp
<code>YES_NO_CANCEL_OPTION</code>	Yes- og No- og Cancel-knapp
<code>OK_CANCEL_OPTION</code>	Yes- og Cancel-knapp

Figur 15.3: Dialogvinduer med metoden `showConfirmDialog()`



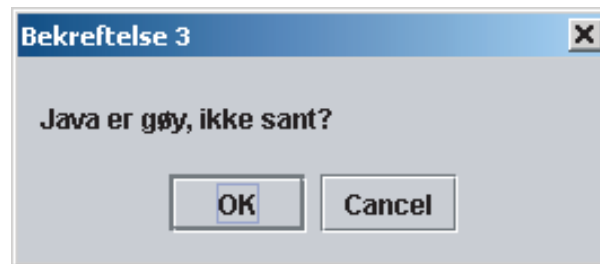
```
JOptionPane.showConfirmDialog( // (1)  
    null, "Skal du gifte deg?");
```

(a)



```
JOptionPane.showConfirmDialog( // (2)  
    null, "Forstår vi hverandre?",  
    "Bekreftelse 2",  
    JOptionPane.YES_NO_OPTION );
```

(b)



```
JOptionPane.showConfirmDialog( // (3)  
    null, "Java er gøy, ikke sant?",  
    "Bekreftelse 3", JOptionPane.OK_CANCEL_OPTION,  
    JOptionPane.PLAIN_MESSAGE);
```

(c)

Program 15.3: Bruk av metoden `showConfirmDialog()`

```
import javax.swing.JOptionPane;

public class BekreftDialog {
    public static void main( String[] args ) {

        int svar1 = JOptionPane.showConfirmDialog( // (1)
            null,                                  // Ingen forelder-ramme
            "Skal du gifte deg?"                  // Ledeteksten
        );                                         // YES-, NO- og CANCEL-knappen

        String svarStr1 = null;
        switch(svar1) {
            case JOptionPane.YES_OPTION:
                svarStr1 = "Gratulerer!"; break;
            case JOptionPane.NO_OPTION:
                svarStr1 = "Nei vel."; break;
            case JOptionPane.CANCEL_OPTION: case JOptionPane.CLOSED_OPTION:
                svarStr1 = "Beklager at jeg spurte."; break;
            default:
                assert false;
        }

        JOptionPane.showMessageDialog(null, svarStr1);
    }
}
```



```

int svar2 = JOptionPane.showConfirmDialog( // (2)
    null,                                // Ingen foreldre-ramme
    "Forstår vi hverandre?",            // Ledeteksten
    "Bekreftelse 2",                    // Tittel på vinduet
    JOptionPane.YES_NO_OPTION           // YES- og NO-knappen
);
String svarStr2 = null;
switch(svar2) {
    case JOptionPane.YES_OPTION:
        svarStr2 = "Bra!"; break;
    case JOptionPane.NO_OPTION: case JOptionPane.CLOSED_OPTION:
        svarStr2 = "Nei vel."; break;
    default:
        assert false;
}

JOptionPane.showMessageDialog(null, svarStr2);

int svar3 = JOptionPane.showConfirmDialog( // (3)
    null,                                // Ingen foreldre-ramme
    "Java er gøy, ikke sant?",          // Ledeteksten
    "Bekreftelse 3",                    // Tittel på vinduet
    JOptionPane.OK_CANCEL_OPTION,       // OK- og CANCEL-knappen
    JOptionPane.PLAIN_MESSAGE           // Beskjedtypen
);

```

```
String svarStr3 = null;
switch(svar3) {
    case JOptionPane.OK_OPTION:
        svarStr3 = "Da er vi enige!"; break;
    case JOptionPane.CANCEL_OPTION: case JOptionPane.CLOSED_OPTION:
        svarStr3 = "Synd at du ikke vil bekrefte."; break;
    default:
        assert false;
}
JOptionPane.showMessageDialog(null, svarStr3);

System.exit(0);
}
```

Støtte for enkel GUI-dialog

- Se filen `GUIDialog.java` (Program 15.4).

Statiske metoder i klassen <code>GUIDialog</code>	Beskrivelse
<code>int lesHelTall(Object ledeTekst)</code>	Leser en <code>int</code> verdi
<code>double lesFlyttall(Object ledeTekst)</code>	Leser en <code>double</code> verdi.
<code>String lesStreng(Object ledeTekst)</code>	Leser en ikke-tom streng.
<code>void skrivMelding(Object melding)</code>	Skriver en melding
<code>int bekreft(Object opplysninger)</code>	Bekrefter opplysninger.

- Program 15.5 viser eksempel på bruk av klassen GUIDialog.

```
public class GUIDialogDemo {  
    public static void main(String[] args) {  
        GUIDialog.skrivMelding(  
            "Du blir bedt om å skrive inn ett heltall om gangen."  
            + " Et negativt tall vil avslutte innlesingen.");  
        int maks = 0;  
        while (true) {  
            int n = GUIDialog.lesHeltall("Skriv et heltall.");  
            if (n < 0)  
                break;  
            if (n > maks)  
                maks = n;  
        }  
        GUIDialog.skrivMelding("Største heltall inntastet: " + maks);  
        System.exit(0);  
    }  
}
```