DIALOGE

Dialoge

von `java.awt.Dialog` abgeleitet

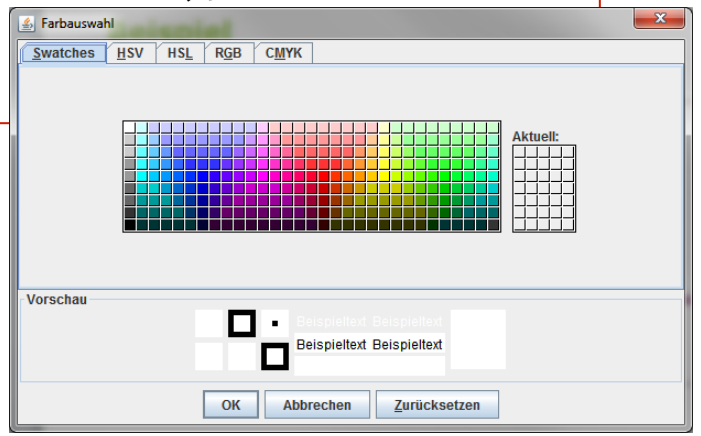
- jeder Dialog ist von einem Fenster abhängig (Elternfenster; owner)
- wenn das Fenster „zerstört “ wird, dann auch die davon abhängigen Dialoge
- **modaler** Dialog: owner ist inaktiv bis Dialog beendet wird
  z.B. `JOptionPane`
- **nicht-modaler** Dialog: owner kann weiterhin benutzt werden
  muss `JDiaolg` sein
- Die Handhabung von `JFrame` und `JDialog` ist sehr ähnlich (aber z.B. nicht maximieren und minimieren; immer ein owner – evtl. null)

Fenster und Dialoge – Übersicht

Klasse	Kurzbeschreibung
`JFrame`	▪ Grundlage für die meisten GUIs ▪ mit Menüleiste
`JDialog`	▪ Dialogfenster ▪ keine Menüleiste ▪ immer abhängig von einem anderen Fenster
`JColorChooser`	▪ Dialog zur Auswahl einer Farbe aus Farbpalette ▪ kann in JDialog oder JFrame integriert werden
`JFileChooser`	▪ Dialog zur Auswahl von Dateien zum Öffnen oder Speichern
`JOptionPane`	▪ Verwendung für Fehlermeldungen, Benutzerbestätigungen, kurze Eingaben usw.

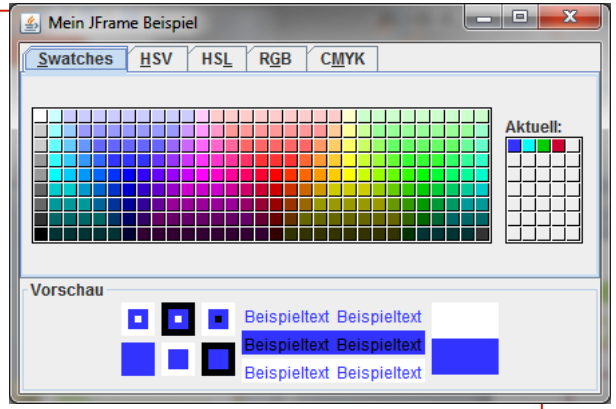
Beispiel
JColorChooser (Methode showDialog)

```
import java.awt.Color;
import javax.swing.JColorChooser;

public class FarbauswahlBeispiel
{
        public static void main(String[] args)
        {
                Color ausgewaehlteFarbe =
        JColorChooser.showDialog(null, "Farbauswahl", null);
                // statische Methode der Klasse JColorChooser
                System.out.println(ausgewaehlteFarbe);
        }
}
```



Beispiel

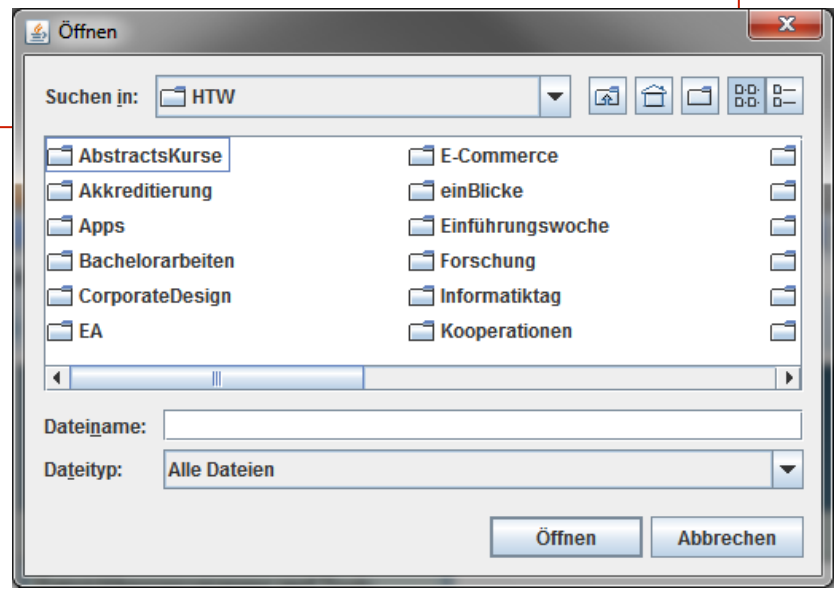
JColorChooser (Objekt erzeugen und in JFrame einbinden)




```
import javax.swing.JColorChooser;
import javax.swing.JFrame;

public class FarbauswahlFrame
{
    public static void main(String[] args)
    {
        JFrame meinJFrame = new JFrame();
        meinJFrame.setTitle("Mein JFrame Beispiel");
        meinJFrame.setSize(450,300);
        JColorChooser colorChooser = new JColorChooser();
        meinJFrame.getContentPane().add(colorChooser);
        // mit add zur aktuellen ContentPane hinzufügen
        meinJFrame.setVisible(true);
    }
}
```

Beispiel
JFileChooser

```
import javax.swing.JFileChooser;

public class DateiauswahlBeispiel1
{
    public static void main(String[] args)
    {
        JFileChooser chooser = new JFileChooser();
        chooser.showOpenDialog(null);
    }
}
```

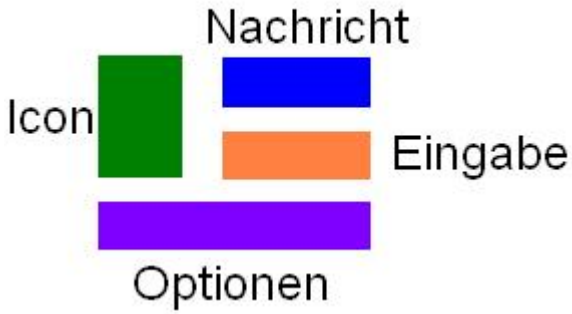


- showOpenDialog öffnet den Öffnen-Dialog
- showSaveDialog öffnet den Speichern-Dialog
- versuchen Sie mal **showDialog(null, "Test")**

Beispiel
JFileChooser

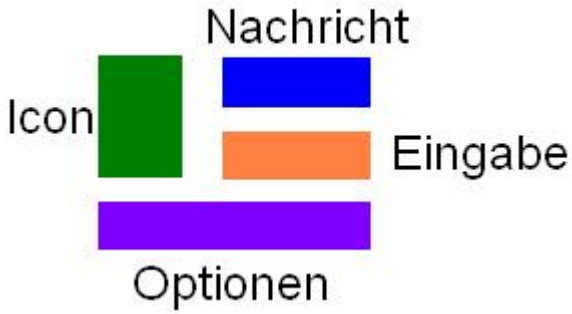
```
import javax.swing.JFileChooser;

public class DateiauswahlBeispiel2
{
    public static void main(String[] args)
    {
        JFileChooser chooser = new JFileChooser();
        int rueckgabeWert = chooser.showOpenDialog(null);

        /* Abfrage, ob auf "Öffnen" geklickt wurde */
        if(rueckgabeWert == JFileChooser.APPROVE_OPTION)
        {
            System.out.println("Die zu öffnende Datei ist: " +
                    chooser.getSelectedFile().getName());
        }
    }
}
```

JOptionPane
Nachrichtentypen



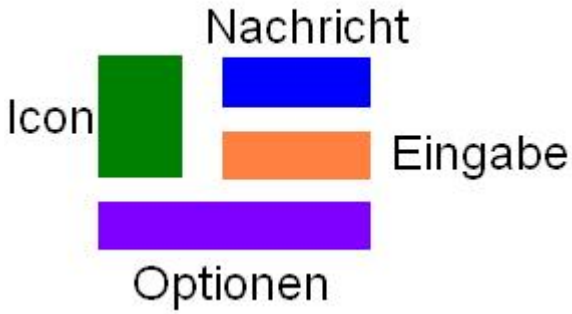
Nachrichtentyp	Kurzbeschreibung
`ERROR_MESSAGE`	▪ Fehlermeldung ▪ rote X als Icon
`INFORMATION_MESSAGE`	▪ Informationsmeldung ▪ kleines i als Icon
`WARNING_MESSAGE`	▪ Warnmeldung ▪ Ausrufezeichen im Warndreieck
`QUESTION_MESSAGE`	▪ Rückfrage an den Nutzer ▪ Fragezeichen als Icon
`PLAIN_MESSAGE`	▪ einfache Textmeldung ▪ kein Icon

JOptionPane
Optionstypen



Optionstyp	Kurzbeschreibung
`DEFAULT_OPTION`	▪ Ok-Button
`YES_NO_OPTION`	▪ Ja- und Nein-Buttons
`YES_NO_CANCEL_OPTION`	▪ Ja-, Nein- und Abbrechen-Buttons
`OK_CANCEL_OPTION`	▪ Ok- und Abbrechen-Buttons

JOptionPane
Methoden zur Erstellung

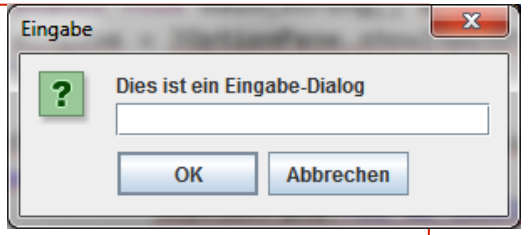


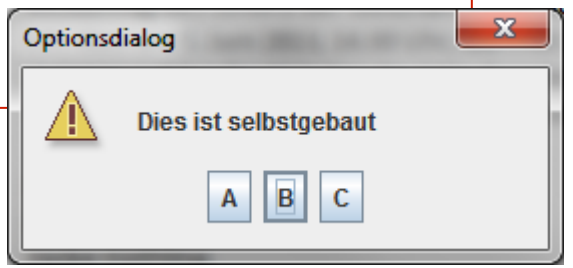
- `static int` **`showConfirmDialog`**`(Component parentComponent, Object message)`
- `static String` **`showInputDialog`**`(Component parentComponent, Object message)`
- `static void` **`showMessageDialog`**`(Component parentComponent, Object message)`
- `static int` **`showOptionDialog`**`(Component parentComponent, Object message, String title, int optionType, int messageType, Icon icon, Object[] options, Object initialValue)`

Konstante	Kurzbeschreibung
YES_OPTION	▪ Ja wurde geklickt (wie Ok)
NO_OPTION	▪ Nein wurde geklickt
CANCEL_OPTION	▪ Abbrechen wurde geklickt
OK_OPTION	▪ Ok wurde geklickt (wie Ja)
CLOSED_OPTION	▪ JOptionPane wurde geschlossen ohne zu klicken

Beispiel
JOptionPane

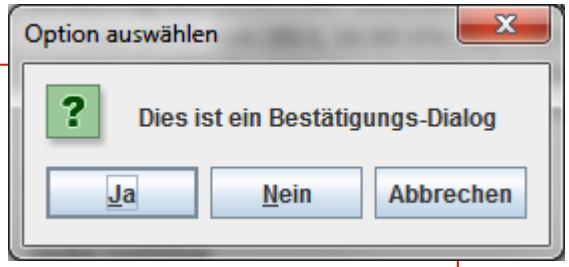
```
import javax.swing.*;

public class OptionPaneBeispiel {

public static void main(String[] args) {
String eingabe = JOptionPane.showInputDialog("Dies ist ein Eingabe-
Dialog");

int bestaetigung = JOptionPane.showOptionDialog(null, "Dies ist
selbstgebaut","Optionsdialog",
                JOptionPane.YES_NO_CANCEL_OPTION,
                JOptionPane.WARNING_MESSAGE, null,
                new String[]{"A", "B", "C"}, "B");
}
}
```

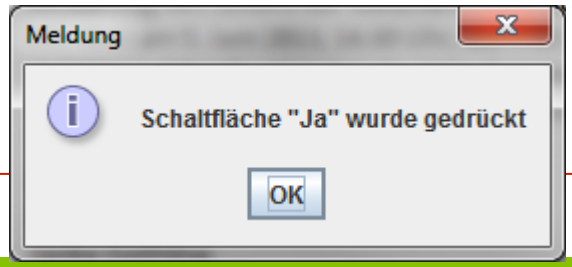




Beispiel
JOptionPane

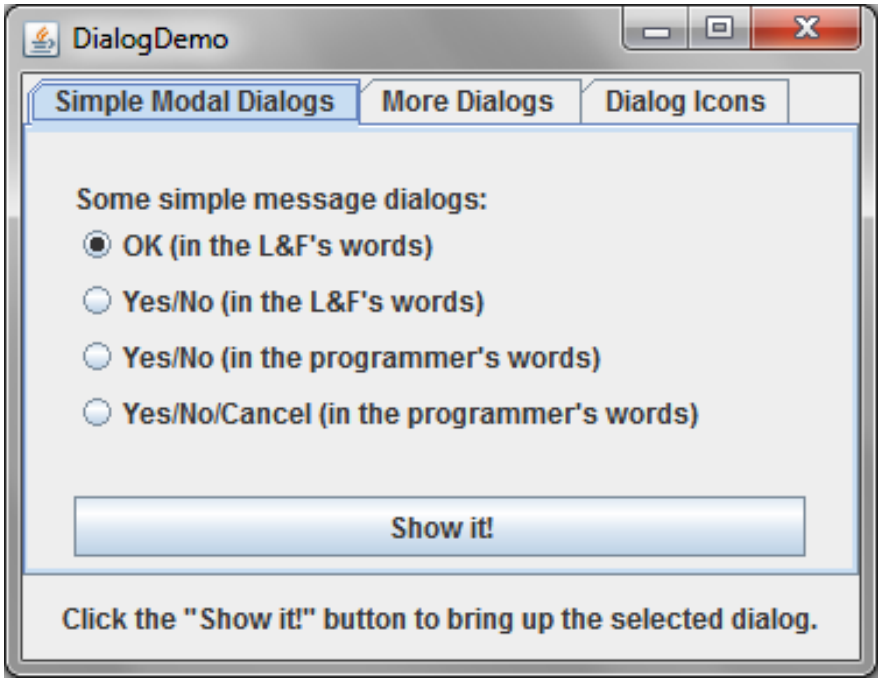
```
import javax.swing.*;

public class OptionPaneBeispiel {

public static void main(String[] args) {
     int bestaetigung = JOptionPane.showConfirmDialog(null, "Dies ist ein
     Bestätigungs-Dialog");

     switch(bestaetigung){
          case JOptionPane.YES_OPTION : JOptionPane.showMessageDialog(null,
          "Schaltfläche \"Ja\" wurde gedrückt"); break;// wie OK_OPTION
          case JOptionPane.NO_OPTION : JOptionPane.showMessageDialog(null,
          "Schaltfläche \"Nein\" wurde gedrückt"); break;
          case JOptionPane.CANCEL_OPTION : JOptionPane.showMessageDialog(null,
          "Schaltfläche \"Abbrechen\" wurde gedrückt"); break;
          case JOptionPane.CLOSED_OPTION : JOptionPane.showMessageDialog(null,
          "geschlossen - nichts geklickt"); break;
     }
}
}
```





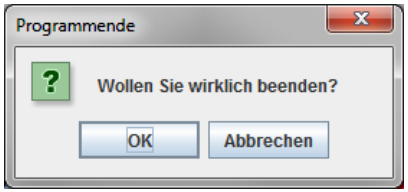
JOptionPane Beispiele

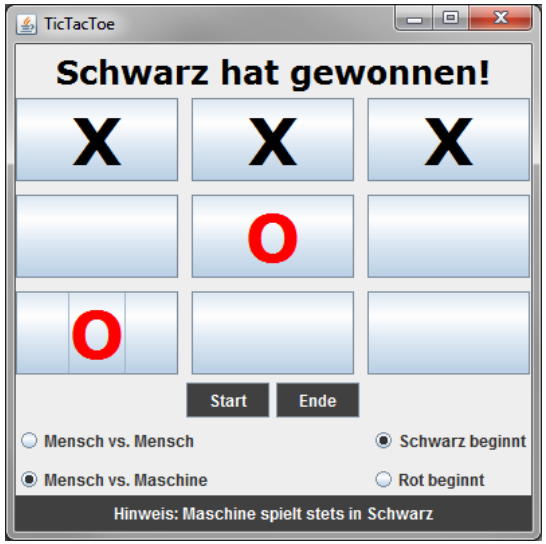
http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html



http://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html

Aufgabe 6





- Erweitern Sie das Spiel TicTacToe!

- mithilfe von RadioButtons (und ButtonGroup) soll eingestellt werden können,

1. ob Mensch gegen Mensch oder Mensch gegen Maschine spielt und
2. ob Schwarz beginnt oder Rot (wenn der Computer spielt, dann stets mit Schwarz)

- bei Drücken des „Start"-Buttons wird das Spiel den Einstellungen der Radiobuttons entsprechend gestartet
- bei Drücken des „Ende"-Buttons erscheint ein Bestätigungsdialog (siehe Bild). Nur wenn in diesem Dialog der „OK"-Button gedrückt wird, wird das Spiel beendet: das Fenster verschwindet und das Programm wird beendet (setVisible(false); dispose(); System.exit(0);) ; bei „Abbrechen" wird weitergespielt

- Die Spielstrategie des Computers muss zunächst wie folgt sein:

1. wenn der Computer in seinem Zug gewinnen kann, dann setzt er sein Kreuz so, dass er auch tatsächlich gewinnt (dann ist das Spiel zu Ende)
2. wenn der Mensch im nächsten Zug gewinnen könnte, dann setzt der Computer sein Kreuz so, dass er den Sieg verhindert (wenn das geht)

- wenn weder 1. noch 2., dann können Sie sich überlegen, wie der Computer setzen sollte (einfachste Lösung: auf das nächste leere Feld)