Klasse 'Evaluator (Teil 1/2)'

```
class Evaluator {
  int nrInRow;
  int dimension;
  Evaluator (int dimension, int nrInRow ) {
   this.nrInRow = nrInRow-1;
   this.dimension = dimension;
  boolean succeeded ( Figure [][] fig, int current ) {
   boolean win = false;
   int i;
    for ( int row=0; (row<dimension) && !win; row++ )
      for { int col=0; (col<dimension) && !win; col++ ) {
        win = true;
        i = 0;
        while ( (col+i<dimension) && (i<=nrInRow) && win ) {
          win = fig[row][col+i].symbol == current;
          i++;
        win &= i>nrInRow;
    for ( int row=0; (row<dimension) && !win; row++ )
      for ( int col=0; (col<dimension) && !win; col++ ) {
        win = true;
        i = 0;
        while ( (row+i<dimension) && (i<=nrInRow) && win ) {
          win = fig[row+i][col].symbol == current;
          i++;
        win &= i>nrInRow;
```

Klasse 'Evaluator (Teil 2/2)'

```
for ( int row=0; (row<dimension) && !win; row++ )
    for ( int col=0; (col<dimension) && !win; col++ ) {
     win = true;
     i = 0:
      while ( (row+i<dimension) && (col+i<dimension) && (i<=nrInRow) && win ) {
        win = fig[row+i][col+i].symbol == current;
        i++;
      win &= i>nrInRow;
  for ( int row=0; (row<dimension) && !win; row++ )
    for (int col=dimension-1; (col>=0) && !win; col-- ) {
     win = true;
     i = 0;
     while ( (row+i<dimension) && (col-i>=0) && (i<=nrInRow) && win ) {
        win = fig[row+i][col-i].symbol == current;
        i++;
     win &= i>nrInRow;
  return win;
boolean undecided ( Figure [][] fig, int noFigure )
 boolean set = true;
 for ( int row=0; (row<dimension) && set; row++ )
   for ( int col=0; (col<dimension) && set; col++ )
     set = fig[row][col].symbol != noFigure;
  return set;
```

Klasse 'Figure'

```
import javax.swing.JButton;

class Figure extends JButton {
   static final char NONE = '\0';
   static final char CROSS = 'X';
   static final char CIRCLE = 'O';
   static char currentSymbol = CROSS;
   char symbol = NONE;

Figure() {
    symbol = NONE;
}

static void toggleSymbol () {
   if ( currentSymbol == CROSS )
        currentSymbol = CIRCLE;
   else
        currentSymbol = CROSS;
}
```

Klasse 'PlayGround (Teil 1/2)'

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class PlayGround extends Frame implements ActionListener {
  private final int HEIGHT;
  private final int WIDTH;
                int dimension;
  private Figure [][] fig;
  private Evaluator eval;
  private boolean sinkDown;
  public PlayGround (int dimension, Evaluator eval, boolean sinkDown) {
    super( "Tic Tac Toe" );
   HEIGHT = dimension*100;
    WIDTH = dimension*100;
    this.dimension = dimension;
    fig = new Figure[dimension][dimension];
   this.eval = eval;
    this.sinkDown = sinkDown;
   setSize( WIDTH, HEIGHT );
   setLayout( new GridLayout( dimension, dimension ) );
   for ( int i=0; i<dimension; i++ )
      for ( int j=0; j<dimension; j++ ) {</pre>
        add( fig[i][j] = new Figure() );
        fig[i][j].addActionListener( this );
   addWindowListener( new WindowAdapter()
                        { public void windowClosing ( WindowEvent e ) {System.exit(0);} } );
```

Klasse 'PlayGround (Teil 2/2)'

```
public void actionPerformed( ActionEvent e ) {
  int row;
 for ( int i=0; i<dimension; i++ )</pre>
    for ( int j=0; j<dimension; j++ )</pre>
      if ( e.getSource() == fig[i][j] ) {
        if ( sinkDown ) {
          row = 0;
          while ( (row<dimension) && (fig[row][j].symbol == Figure.NONE) )
            row++;
          row--;
          if ((row>=0) && (fig[row][j].symbol == Figure.NONE) )
            setFigure( row, j );
        } else if ( fig[i][j].symbol == Figure.NONE ) {
          setFigure( i, j );
} // actionPerformed
private void setFigure( int i, int j ) {
  fig[i][j].symbol = Figure.currentSymbol;
 fig[i][j].setText( new Character(Figure.currentSymbol).toString() );
 if ( eval.succeeded( fig, Figure.currentSymbol ) ) {
    for ( int i1=0; i1<dimension; i1++ )
      for ( int j1=0; j1<dimension; j1++ )
        fig[i1][j1].removeActionListener( this );
    new SuccessFrame (Figure.currentSymbol, this);
  } else if ( eval.undecided( fig, Figure.NONE ) ) {
    new UndecidedFrame (this);
  Figure.toggleSymbol();
```

Klasse 'Run'

Klasse 'SuccessFrame'

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class SuccessFrame extends Frame implements ActionListener {
  private final int WIDTH = 200;
  private final int HEIGHT = 150;
  private JButton stop = new JButton( "OK" );
  private Frame playGround;
  public SuccessFrame( char successNr, Frame playGround ) {
    super( "Gewonnen!" );
   this.playGround = playGround;
    setSize( WIDTH, HEIGHT );
    setLayout( new GridLayout( 2, 1 ) );
   Label lb = new Label( "Spieler '"+successNr+"' hat gewonnen!" );
    lb.setAlignment(Label.CENTER);
    add(lb);
    add(stop);
   stop.addActionListener( this );
    show();
  public void actionPerformed( ActionEvent e ) {
   hide();
   playGround.hide();
    System.exit( 0 );
```

Klasse 'UndecidedFrame'

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class UndecidedFrame extends Frame implements ActionListener {
  private final int WIDTH = 300;
  private final int HEIGHT = 150;
  private JButton stop = new JButton( "OK" );
  private Frame playGround;
  public UndecidedFrame( Frame playGround ) {
    super( "Unentschieden!" );
    this.playGround = playGround;
    setSize( WIDTH, HEIGHT );
    setLayout( new GridLayout( 2, 1 ) );
    Label lb = new Label( "Das Spiel endete unentschieden!" );
    lb.setAlignment(Label.CENTER);
    add(lb);
    add(stop);
    stop.addActionListener( this );
    show();
  public void actionPerformed( ActionEvent e ) {
    hide();
    playGround.hide();
    System.exit( 0 );
```