

# Contents

<b>1</b>	<b>Package</b>	<b>2</b>
1.1	Class Awesome . . . . .	2
1.1.1	Declaration . . . . .	2
1.1.2	Constructor summary . . . . .	2
1.1.3	Method summary . . . . .	2
1.1.4	Constructors . . . . .	2
1.1.5	Methods . . . . .	2
<b>2</b>	<b>Package control</b>	<b>4</b>
2.1	Class Spiel . . . . .	4
2.1.1	Declaration . . . . .	4
2.1.2	Field summary . . . . .	4
2.1.3	Constructor summary . . . . .	4
2.1.4	Method summary . . . . .	5
2.1.5	Fields . . . . .	5
2.1.6	Constructors . . . . .	5
2.1.7	Methods . . . . .	6
2.2	Class Spiel.ButtonListener . . . . .	6
2.2.1	Declaration . . . . .	7
2.2.2	Constructor summary . . . . .	7
2.2.3	Method summary . . . . .	7
2.2.4	Constructors . . . . .	7
2.2.5	Methods . . . . .	7
2.3	Class Spiel.RestartListener . . . . .	7
2.3.1	Declaration . . . . .	7
2.3.2	Constructor summary . . . . .	7
2.3.3	Method summary . . . . .	7
2.3.4	Constructors . . . . .	7
2.3.5	Methods . . . . .	8
<b>3</b>	<b>Package data</b>	<b>9</b>
3.1	Class Feld . . . . .	9
3.1.1	Declaration . . . . .	9
3.1.2	Field summary . . . . .	9
3.1.3	Constructor summary . . . . .	9
3.1.4	Method summary . . . . .	10

3.1.5	Fields	10
3.1.6	Constructors	10
3.1.7	Methods	10
3.2	Class Kasten	12
3.2.1	Declaration	12
3.2.2	Field summary	12
3.2.3	Constructor summary	12
3.2.4	Method summary	13
3.2.5	Fields	13
3.2.6	Constructors	13
3.2.7	Methods	13
3.3	Class Pin	14
3.3.1	Declaration	15
3.3.2	Field summary	15
3.3.3	Constructor summary	15
3.3.4	Method summary	15
3.3.5	Fields	15
3.3.6	Constructors	15
3.3.7	Methods	15
3.4	Class Spieler	15
3.4.1	Declaration	16
3.4.2	Field summary	16
3.4.3	Constructor summary	16
3.4.4	Method summary	16
3.4.5	Fields	16
3.4.6	Constructors	17
3.4.7	Methods	17
<b>4</b>	<b>Package test</b>	<b>21</b>
4.1	Class KastGew	21
4.1.1	Declaration	21
4.1.2	Constructor summary	21
4.1.3	Method summary	21
4.1.4	Constructors	21
4.1.5	Methods	22
4.2	Class namen	22
4.2.1	Declaration	22
4.2.2	Field summary	22
4.2.3	Constructor summary	22
4.2.4	Method summary	22
4.2.5	Fields	22
4.2.6	Constructors	22
4.2.7	Methods	23
4.3	Class normWuerfeln	23
4.3.1	Declaration	23
4.3.2	Constructor summary	23

4.3.3	Method summary . . . . .	23
4.3.4	Constructors . . . . .	23
4.3.5	Methods . . . . .	23
4.4	Class <code>wuerfel12</code> . . . . .	23
4.4.1	Declaration . . . . .	23
4.4.2	Constructor summary . . . . .	24
4.4.3	Method summary . . . . .	24
4.4.4	Constructors . . . . .	24
4.4.5	Methods . . . . .	24
4.5	Class <code>wuerfel2</code> . . . . .	24
4.5.1	Declaration . . . . .	24
4.5.2	Constructor summary . . . . .	24
4.5.3	Method summary . . . . .	24
4.5.4	Constructors . . . . .	24
4.5.5	Methods . . . . .	24
<b>5</b>	<b>Package view</b>	<b>25</b>
5.1	Interface <code>ContractView</code> . . . . .	25
5.1.1	Declaration . . . . .	25
5.1.2	All known subinterfaces . . . . .	26
5.1.3	All classes known to implement interface . . . . .	26
5.1.4	Method summary . . . . .	26
5.1.5	Methods . . . . .	26
5.2	Class <code>CustomButton</code> . . . . .	29
5.2.1	Declaration . . . . .	29
5.2.2	Field summary . . . . .	29
5.2.3	Constructor summary . . . . .	29
5.2.4	Method summary . . . . .	29
5.2.5	Fields . . . . .	29
5.2.6	Constructors . . . . .	30
5.2.7	Methods . . . . .	30
5.2.8	Members inherited from class <code>JButton</code> . . . . .	31
5.2.9	Members inherited from class <code>AbstractButton</code> . . . . .	32
5.2.10	Members inherited from class <code>JComponent</code> . . . . .	35
5.2.11	Members inherited from class <code>Container</code> . . . . .	40
5.2.12	Members inherited from class <code>Component</code> . . . . .	43
5.3	Class <code>NameDialog</code> . . . . .	51
5.3.1	Declaration . . . . .	51
5.3.2	Field summary . . . . .	51
5.3.3	Constructor summary . . . . .	52
5.3.4	Method summary . . . . .	52
5.3.5	Fields . . . . .	52
5.3.6	Constructors . . . . .	52
5.3.7	Methods . . . . .	53
5.4	Class <code>NameDialog.EnterKeyListener</code> . . . . .	53
5.4.1	Declaration . . . . .	53

5.4.2	Field summary . . . . .	53
5.4.3	Constructor summary . . . . .	53
5.4.4	Method summary . . . . .	54
5.4.5	Fields . . . . .	54
5.4.6	Constructors . . . . .	54
5.4.7	Methods . . . . .	54
5.4.8	Members inherited from class <code>KeyAdapter</code> . . . . .	54
5.5	Class <code>View</code> . . . . .	54
5.5.1	Declaration . . . . .	54
5.5.2	Field summary . . . . .	55
5.5.3	Constructor summary . . . . .	55
5.5.4	Method summary . . . . .	55
5.5.5	Fields . . . . .	55
5.5.6	Constructors . . . . .	56
5.5.7	Methods . . . . .	56
5.6	Class <code>View.PaintingComponent</code> . . . . .	60
5.6.1	Declaration . . . . .	60
5.6.2	Field summary . . . . .	60
5.6.3	Constructor summary . . . . .	60
5.6.4	Method summary . . . . .	60
5.6.5	Fields . . . . .	61
5.6.6	Constructors . . . . .	61
5.6.7	Methods . . . . .	61
5.6.8	Members inherited from class <code>JComponent</code> . . . . .	62
5.6.9	Members inherited from class <code>Container</code> . . . . .	67
5.6.10	Members inherited from class <code>Component</code> . . . . .	70
5.7	Class <code>WinDialog</code> . . . . .	78
5.7.1	Declaration . . . . .	78
5.7.2	Field summary . . . . .	78
5.7.3	Constructor summary . . . . .	79
5.7.4	Method summary . . . . .	79
5.7.5	Fields . . . . .	79
5.7.6	Constructors . . . . .	79
5.7.7	Methods . . . . .	79
5.8	Class <code>WinDialog.EnterKeyListener</code> . . . . .	80
5.8.1	Declaration . . . . .	80
5.8.2	Constructor summary . . . . .	80
5.8.3	Method summary . . . . .	80
5.8.4	Constructors . . . . .	80
5.8.5	Methods . . . . .	80
5.8.6	Members inherited from class <code>KeyAdapter</code> . . . . .	80

# Chapter 1

## Package

<i>Package Contents</i>	<i>Page</i>
Classes	
<b>Awesome</b> .....	<a href="#">2</a>
Hauptklasse des Spiels.	

### 1.1 Class Awesome

Hauptklasse des Spiels.

#### 1.1.1 Declaration

```
public class Awesome
    extends java.lang.Object
```

#### 1.1.2 Constructor summary

    Awesome()

#### 1.1.3 Method summary

    main(String[]) Startet das Spiel.

#### 1.1.4 Constructors

- Awesome

```
    public Awesome()
```

#### 1.1.5 Methods

- main

```
public static void main(java.lang.String[] args)
```

– **Description**

Startet das Spiel.

– **Parameters**

\* **args** – Argumente der Kommandozeile beim Aufruf. Nicht benoetigt.

## Chapter 2

# Package control

<i>Package Contents</i>	<i>Page</i>
<b>Classes</b>	
<b>Spiel</b> .....	<a href="#">4</a>
Das Spiel selbst	
<b>Spiel.ButtonListener</b> .....	<a href="#">6</a>
Der Listener fuer alle Buttons die zum Spielfeld gehoeren.	
<b>Spiel.RestartListener</b> .....	<a href="#">7</a>
Der Listener fuer einen Button mit dem das Spiel neu gestartet werden kann.	

### 2.1 Class Spiel

Das Spiel selbst

#### 2.1.1 Declaration

```
public class Spiel
    extends java.lang.Object
```

#### 2.1.2 Field summary

**activePlayer** Fuer den Zug aktiver Spieler.  
**isRunning** true solange das Spiel laeuft.  
**kaesten** Spielfeld mit 9 Kaesten.  
**kaestenGewonnen** Zaehlt, wieviele Kaesten gewonnen wurden  
**spieler** Die Spieler eines Spiels.  
**view** Speichert die View.

#### 2.1.3 Constructor summary

**Spiel(View)** Erstellt ein Spiel mit einem Spielfeld.

### 2.1.4 Method summary

- gameEnd()** Füllt alle Buttons mit der Farbe des Gewinners und ruft ein Dialogfenster auf.
- gameStart()** Bestimmt zufaellig, welcher Spieler beginnen soll und startet danach den ersten Zug.
- initGame()** Initialisiert das Spielfeld, zeigt danach das Hauptfenster an und startet das Spiel.
- restartGame()** Startet das Spiel neu.
- startRound(Kasten[])** Startet den Zug, wuerfelt fuer den aktuellen Spieler und bietet ihm die moeglichen Felder an.

### 2.1.5 Fields

- **private data.Kasten[] kaesten**
  - Spielfeld mit 9 Kaesten.
- **private final data.Spieler[] spieler**
  - Die Spieler eines Spiels.
- **private data.Spieler activePlayer**
  - Fuer den Zug aktiver Spieler.
- **private view.View view**
  - Speichert die View.
- **private boolean isRunning**
  - true solange das Spiel laeuft.
- **private int kaestenGewonnen**
  - Zaehlt, wieviele Kaesten gewonnen wurden

### 2.1.6 Constructors

- **Spiel**

**public Spiel(view.View view)**

- **Description**  
Erstellt ein Spiel mit einem Spielfeld.



### 2.1.7 Methods

- **gameEnd**

```
private void gameEnd()
```

- **Description**

Fuellt alle Buttons mit der Farbe des Gewinners und ruft ein Dialogfenster auf.

- **gameStart**

```
private void gameStart()
```

- **Description**

Bestimmt zufaellig, welcher Spieler beginnen soll und startet danach den ersten Zug.

- **initGame**

```
private void initGame()
```

- **Description**

Initialisiert das Spielfeld, zeigt danach das Hauptfenster an und startet das Spiel.

- **restartGame**

```
private void restartGame()
```

- **Description**

Startet das Spiel neu.

- **startRound**

```
private void startRound(data.Kasten[] kaesten)
```

- **Description**

Startet den Zug, wuerfelt fuer den aktuellen Spieler und bietet ihm die moeglichen Felder an.

- **Parameters**

\* **kaesten** – - Das Spielfeld

## 2.2 Class Spiel.ButtonListener

Der Listener fuer alle Buttons die zum Spielfeld gehoeren.

### 2.2.1 Declaration

```
class Spiel.ButtonListener
extends java.lang.Object implements java.awt.event.ActionListener
```

### 2.2.2 Constructor summary

```
Spiel.ButtonListener()
```

### 2.2.3 Method summary

```
actionPerformed(ActionEvent)
```

### 2.2.4 Constructors

- **Spiel.ButtonListener**

```
Spiel.ButtonListener()
```

### 2.2.5 Methods

- **actionPerformed**

```
void actionPerformed(java.awt.event.ActionEvent arg0)
```

## 2.3 Class Spiel.RestartListener

Der Listener fuer einen Button mit dem das Spiel neu gestartet werden kann.

### 2.3.1 Declaration

```
class Spiel.RestartListener
extends java.lang.Object implements java.awt.event.ActionListener
```

### 2.3.2 Constructor summary

```
Spiel.RestartListener()
```

### 2.3.3 Method summary

```
actionPerformed(ActionEvent)
```

### 2.3.4 Constructors

- **Spiel.RestartListener**

```
Spiel.RestartListener()
```

### 2.3.5 Methods

- **actionPerformed**

```
void actionPerformed(java.awt.event.ActionEvent arg0)
```

## Chapter 3

# Package data

<i>Package Contents</i>	<i>Page</i>
<b>Classes</b>	
<b>Feld</b> .....	<a href="#">9</a>
Ein Feld, dass der Spieler gewinnnen kann.	
<b>Kasten</b> .....	<a href="#">12</a>
Kasten, der gewonnen werden kann.	
<b>Pin</b> .....	<a href="#">14</a>
Ein Pin, der einen Spieler halten kann.	
<b>Spieler</b> .....	<a href="#">15</a>
Der Spieler	

### 3.1 Class Feld

Ein Feld, dass der Spieler gewinnnen kann. Drei Felder in Reihe lassen einen Kasten gewinnen.

#### 3.1.1 Declaration

```
public class Feld
    extends java.lang.Object
```

#### 3.1.2 Field summary

**feldIndex** Indes des Feldes.  
**feldNummer** Spiel- / Wuerfelnummer des Feldes.  
**highlight** Markierung als verfuegbar zum setzen wenn true.  
**highlightToDelete** Markierung als verfuegbar zum loeschen wenn true.  
**pin** Pin der in dem Feld gesetzt wird.

#### 3.1.3 Constructor summary

**Feld(int, int)** Erstellt ein Feld.

### 3.1.4 Method summary

**getFeldIndex()** Gibt den Index des Feldes zurueck.  
**getFeldNummer()** Gibt die Spielnummer des Feldes zurueck.  
**getHighlight()** Gibt zurueck ob ein Feld besetzbar ist.  
**getHighlightToDelete()** Gibt zurueck ob ein Feld loeschbar ist.  
**getPin()** Gibt den Pin des Feldes zurueck.  
**setHighlight(boolean)** Setzt ein Feld als besetzbar.  
**setHighlightToDelete(boolean)** Setzt ein Feld als loeschbar.  
**setPin(Pin)** Setzt den Pin des Feldes.

### 3.1.5 Fields

- **private Pin pin**
  - Pin der in dem Feld gesetzt wird.
- **private int feldNummer**
  - Spiel- / Wuerfelnummer des Feldes.
- **private int feldIndex**
  - Indes des Feldes.
- **private boolean highlight**
  - Markierung als verfuegbar zum setzen wenn true.
- **private boolean highlightToDelete**
  - Markierung als verfuegbar zum loeschen wenn true.

### 3.1.6 Constructors

- **Feld**

```
public Feld(int feldNummer,int feldIndex)
```

- **Description**  
Erstellt ein Feld.
- **Parameters**
  - \* **feldNummer** – Spielnummer des Feldes. Nicht null.
  - \* **feldIndex** – Index des Feldes. Nicht null.

### 3.1.7 Methods

- **getFeldIndex**

```
public int getFeldIndex()
```

- **Description**  
Gibt den Index des Feldes zurueck.
- **Returns** – Index des Feldes.

- **getFeldNummer**

```
public int getFeldNummer()
```

- **Description**  
Gibt die Spielnummer des Feldes zurueck.
- **Returns** – Spielnummer des Feldes.

- **getHighlight**

```
public boolean getHighlight()
```

- **Description**  
Gibt zurueck ob ein Feld besetzbar ist.
- **Returns** – true wenn besetzbar.

- **getHighlightToDelete**

```
public boolean getHighlightToDelete()
```

- **Description**  
Gibt zurueck ob ein Feld loeschbar ist.
- **Returns** – true wenn loeschbar.

- **getPin**

```
public Pin getPin()
```

- **Description**  
Gibt den Pin des Feldes zurueck.
- **Returns** – Gibt den Pin zurueck.

- **setHighlight**

```
public void setHighlight(boolean highlight)
```

- **Description**  
Setzt ein Feld als besetzbar.

- **Parameters**

- \* **highlight** – true wenn besetzbar.

- **setHighlightToDelete**

```
public void setHighlightToDelete(boolean highlightToDelete)
```

- **Description**

- Setzt ein Feld als loeschbar.

- **Parameters**

- \* **highlightToDelete** – true wenn loeschbar.

- **setPin**

```
public void setPin(Pin pin)
```

- **Description**

- Setzt den Pin des Feldes.

- **Parameters**

- \* **pin** – Zu setzender Pin.

## 3.2 Class Kasten

Kasten, der gewonnen werden kann. Drei Kaesten in Reihe fuer einen Spieler gewinnen das Spiel.

### 3.2.1 Declaration

```
public class Kasten
  extends java.lang.Object
```

### 3.2.2 Field summary

**felder** 9 Felder pro Kasten.

**kastenNummer** Die ID-Nummer des Kastens zur leichteren Adressierung.

**spieler** Diese Variable wird gesetzt sobald ein Spieler einen Kasten fr sich entschieden hat.

### 3.2.3 Constructor summary

**Kasten(int)** Erstellt einen Kasten.

### 3.2.4 Method summary

**erzeugeFeld()** Erzeugt ein `Feld[]` mit Zufallszahlen und gibt es zurueck.  
**getFelder()** Gibt die beinhalteten Felder des Kastens zurueck.  
**getKastenNummer()** Gibt die Kastennummer zurueck.  
**getSpieler()** Gibt den Spieler zurueck welchem der Kasten gehoert.  
**setFelder(Feld[])** Setzt die Felder eines Kastens.  
**setSpieler(Spieler)** Setzt den Spieler der Kasten gewonnen hat.

### 3.2.5 Fields

- **private Feld[] felder**
  - 9 Felder pro Kasten.
- **private int kastenNummer**
  - Die ID-Nummer des Kastens zur leichteren Adressierung.
- **private Spieler spieler**
  - Diese Variable wird gesetzt sobald ein Spieler einen Kasten fr sich entschieden hat.

### 3.2.6 Constructors

- **Kasten**

```
public Kasten(int kastenNummer)
```

- **Description**  
Erstellt einen Kasten.
- **Parameters**
  - \* **kastenNummer** – Nummer des Kastens im Spiel.

### 3.2.7 Methods

- **erzeugeFeld**

```
public Feld[] erzeugeFeld()
```

- **Description**  
Erzeugt ein `Feld[]` mit Zufallszahlen und gibt es zurueck.

- **getFelder**

```
public Feld[] getFelder()
```



- **Description**  
Gibt die beinhalteten Felder des Kastens zurueck.
- **Returns** – Gibt die Felder zurueck.

- **getKastenNummer**

```
public int getKastenNummer()
```

- **Description**  
Gibt die Kastennummer zurueck.
- **Returns** – Die Kastennummer.

- **getSpieler**

```
public Spieler getSpieler()
```

- **Description**  
Gibt den Spieler zurueck welchem der Kasten gehoert.
- **Returns** – Spieler, dem der Kasten gehoert.

- **setFelder**

```
public void setFelder(Feld[] felder)
```

- **Description**  
Setzt die Felder eines Kastens.
- **Parameters**
  - \* **felder** – Felder gesetzt werden sollen. Muessen genau neun (9) sein.

- **setSpieler**

```
public void setSpieler(Spieler spieler)
```

- **Description**  
Setzt den Spieler der Kasten gewonnen hat.
- **Parameters**
  - \* **spieler** – Der Spieler, der den Kasten gewonnen hat.

### 3.3 Class Pin

Ein Pin, der einen Spieler halten kann.

### 3.3.1 Declaration

```
public class Pin
    extends java.lang.Object
```

### 3.3.2 Field summary

**spieler** Spieler dem der Pin gehoert.

### 3.3.3 Constructor summary

**Pin(Spieler)** Erstellt einen Pin fuer einen Spieler.

### 3.3.4 Method summary

**getSpieler()** Gibt den Spieler des Pin zurueck.

### 3.3.5 Fields

- **private Spieler spieler**
  - Spieler dem der Pin gehoert.

### 3.3.6 Constructors

- **Pin**

```
public Pin(Spieler spieler)
```

- **Description**  
Erstellt einen Pin fuer einen Spieler.
- **Parameters**
  - \* **spieler** – Spieler dem der Pin gehoert. nicht null.

### 3.3.7 Methods

- **getSpieler**

```
public Spieler getSpieler()
```

- **Description**  
Gibt den Spieler des Pin zurueck.
- **Returns** – Spieler des Pin.

## 3.4 Class Spieler

Der Spieler

### 3.4.1 Declaration

```
public class Spieler
    extends java.lang.Object
```

### 3.4.2 Field summary

**farbe** Farbe des Spielers.  
**name** Name des Spielers.  
**punkte** Erzielte Punkte.  
**wuerfelErgebnis** Aktuell gewuerfelte Zahl.

### 3.4.3 Constructor summary

**Spieler(String, Color)** Erstellt einen Spieler.

### 3.4.4 Method summary

**bieteFelderAn(Kasten[])** Die Methode vergleicht alle Felder mit dem Wuerfel-  
 ergebnis und gibt die Indizes der Kaesten und der dazugehrigen Felder zurueck.  
**bieteLoeschFelderAn(Kasten[])** Markiert die loeschbaren Felder des Gegners.  
**checkList(Kasten, ArrayList, int, int, int)** Ausgelagerte Hilfsmethode fuer die  
 Methode pruefeKasten().  
**erhoehePunkte()** Erhoeht die Punkte um 1.  
**getColor()** Gibt die Farbe des Spielers zurueck.  
**getName()** Gibt den Namen des Spielers zurueck.  
**getPunkte()** Gibt die erspielten Punkte zurueck.  
**getWuerfelErgebnis()** Gibt das Wurfelergebnis zurueck.  
**pinLoeschen(Kasten[], int, int)** Loescht einen Pin  
**pinSetzen(Kasten[], int, int)** Setzt einen Pin auf ein freies Feld.  
**pruefeKasten(Kasten)** Prueft, ob der Kasten gewonnen wurde  
**pruefeSpielfeld(Kasten[])** Prueft ob der Spieler das Spiel gewonnen hat.  
**setWuerfelErgebnis(int)** Setter zu Testzwecken.  
**verringerePunkte()** Verringert die Punkte um 1.  
**wuerfeln()** Wuerfelt fuer den Spieler eine Zufallszahl und speichert diese in der  
 Instanzvariable wuerfelErgebnis.

### 3.4.5 Fields

- **private final java.lang.String name**  
 – Name des Spielers.
- **private int punkte**  
 – Erzielte Punkte.
- **private int wuerfelErgebnis**  
 – Aktuell gewuerfelte Zahl.
- **private final java.awt.Color farbe**

- Farbe des Spielers.

### 3.4.6 Constructors

- **Spieler**

```
public Spieler(java.lang.String name,java.awt.Color farbe)
```

- **Description**  
Erstellt einen Spieler.
- **Parameters**
  - \* **name** – Name des Spielers. Nicht null.
  - \* **farbe** – Farbe des Spielers. Nicht null.

### 3.4.7 Methods

- **bieteFelderAn**

```
public java.util.ArrayList bieteFelderAn(Kasten[] kaesten)
```

- **Description**  
Die Methode vergleicht alle Felder mit dem Wuerfelergebnis und gibt die Indizes der Kaesten und der dazugehorigen Felder zurueck.
- **Parameters**
  - \* **kaesten** – Das Spielfeld
- **Returns** – String-ArrayList mit den Indizes der freien Felder

- **bieteLoeschFelderAn**

```
public void bieteLoeschFelderAn(Kasten[] kaesten)
```

- **Description**  
Markiert die loeschbaren Felder des Gegners.
- **Parameters**
  - \* **kaesten** – Das Spielfeld.

- **checkList**

```
private boolean checkList(Kasten k,java.util.ArrayList  
pruefListe,int x1,int x2,int x3)
```

- **Description**

Ausgelagerte Hilfsmethode fuer die Methode `pruefeKasten()`. Prueft anhand der uebergebenen Indizes, ob ein Kasten gewonnen wurde.

- **Parameters**

- \* `k` – Der Kasten, der geprueft werden soll.
- \* `pruefListe` – Die Liste mit den Indizes der Feld, die der Spieler gewonnen hat.
- \* `x1` – Index 1
- \* `x2` – Index 2
- \* `x3` – Index 3

- **Returns** – true, wenn der Kasten gewonnen wurde. Sonst false.

- **erhoehePunkte**

```
public void erhoehePunkte()
```

- **Description**

Erhoeht die Punkte um 1.

- **getColor**

```
public java.awt.Color getColor()
```

- **Description**

Gibt die Farbe des Spielers zurueck.

- **Returns** – Die Farbe des Spielers.

- **getName**

```
public java.lang.String getName()
```

- **Description**

Gibt den Namen des Spielers zurueck.

- **Returns** – Der Name des Spielers.

- **getPunkte**

```
public int getPunkte()
```

- **Description**

Gibt die erspielten Punkte zurueck.

- **Returns** – Punkteanzahl.

- **getWuerfelErgebnis**

```
public int getWuerfelErgebnis()
```

- **Description**

- Gibt das Wuerfelergebnis zurueck.

- **Returns** – Wuerfelergebnis.

- **pinLoeschen**

```
public void pinLoeschen(Kasten[] kaesten, int feldIndex, int  
kastenIndex)
```

- **Description**

- Loescht einen Pin

- **Parameters**

- \* **kaesten** – Das Spielfeld

- \* **kastenIndex** – Der Kastenindex des gewaehlten Feldes

- \* **feldIndex** – Der Feldindex des gewaehlten Feldes

- **pinSetzen**

```
public boolean pinSetzen(Kasten[] kaesten, int feldIndex, int  
kastenIndex)
```

- **Description**

- Setzt einen Pin auf ein freies Feld.

- **Parameters**

- \* **kaesten** – Das Spielfeld

- \* **feldIndex** – Der Feldindex des gewaehlten Feldes

- \* **kastenIndex** – Der Kastenindex des gewaehlten Feldes

- **Returns** – Gibt true zurueck, wenn das Spiel gewonnen wurde. Sonst false.

- **pruefeKasten**

```
private boolean pruefeKasten(Kasten k)
```

- **Description**

- Prueft, ob der Kasten gewonnen wurde

- **Parameters**

- \* **k** – zu pruefender Kasten

- **Returns** – true, wenn der Kasten gewonnen wurde. Sonst false.

- **pruefeSpielfeld**

```
private boolean pruefeSpielfeld(Kasten[] kaesten)
```

- **Description**  
Prueft ob der Spieler das Spiel gewonnen hat.
- **Parameters**
  - \* **kaesten** – Das Spielfeld. Nicht null.
- **Returns** – true wenn gewonnen. Sonst false.

- **setWuerfelErgebnis**

```
public void setWuerfelErgebnis(int wuerfelErgebnis)
```

- **Description**  
Setter zu Testzwecken.
- **Parameters**
  - \* **wuerfelErgebnis** – Zu setzendes Ergebnis.

- **verringerePunkte**

```
public void verringerePunkte()
```

- **Description**  
Verringert die Punkte um 1.

- **wuerfeln**

```
public int wuerfeln()
```

- **Description**  
Wuerfelt fuer den Spieler eine Zufallszahl und speichert diese in der Instanzvariable wuerfelErgebnis.
- **Returns** – Das Wuerfelergebnis als Integer

## Chapter 4

# Package test

<i>Package Contents</i>	<i>Page</i>
<b>Classes</b>	
<b>KastGew</b> .....	<a href="#">21</a>
<b>namen</b> .....	<a href="#">22</a>
<b>normWuerfeln</b> .....	<a href="#">23</a>
<b>wuerfel12</b> .....	<a href="#">23</a>
<b>wuerfel2</b> .....	<a href="#">24</a>

### 4.1 Class KastGew

#### 4.1.1 Declaration

```
public class KastGew
    extends java.lang.Object
```

#### 4.1.2 Constructor summary

```
    KastGew()
```

#### 4.1.3 Method summary

```
    main(String[])
```

#### 4.1.4 Constructors

- **KastGew**

```
    public KastGew()
```



### 4.1.5 Methods

- **main**

```
public static void main(java.lang.String[] args)
```

## 4.2 Class namen

### 4.2.1 Declaration

```
public class namen  
    extends java.lang.Object
```

### 4.2.2 Field summary

```
    farbe  
    farbe2  
    name  
    name2
```

### 4.2.3 Constructor summary

```
    namen()
```

### 4.2.4 Method summary

```
    main(String[])
```

### 4.2.5 Fields

- java.lang.String **name**
- java.lang.String **name2**
- java.awt.Color **farbe**
- java.awt.Color **farbe2**

### 4.2.6 Constructors

- **namen**

```
    public namen()
```

### 4.2.7 Methods

- **main**

```
public static void main(java.lang.String[] args)
```

## 4.3 Class normWuerfeln

### 4.3.1 Declaration

```
public class normWuerfeln  
    extends java.lang.Object
```

### 4.3.2 Constructor summary

```
normWuerfeln()
```

### 4.3.3 Method summary

```
input(String)  
main(String[])
```

### 4.3.4 Constructors

- **normWuerfeln**

```
public normWuerfeln()
```

### 4.3.5 Methods

- **input**

```
public static java.lang.String input(java.lang.String ausgabe)
```

- **main**

```
public static void main(java.lang.String[] args)
```

## 4.4 Class wuerfel12

### 4.4.1 Declaration

```
public class wuerfel12  
    extends java.lang.Object
```

#### 4.4.2 Constructor summary

```
wuerfel12()
```

#### 4.4.3 Method summary

```
main(String[])
```

#### 4.4.4 Constructors

- **wuerfel12**

```
public wuerfel12()
```

#### 4.4.5 Methods

- **main**

```
public static void main(java.lang.String[] args)
```

### 4.5 Class wuerfel2

#### 4.5.1 Declaration

```
public class wuerfel2  
    extends java.lang.Object
```

#### 4.5.2 Constructor summary

```
wuerfel2()
```

#### 4.5.3 Method summary

```
main(String[])
```

#### 4.5.4 Constructors

- **wuerfel2**

```
public wuerfel2()
```

#### 4.5.5 Methods

- **main**

```
public static void main(java.lang.String[] args)
```

## Chapter 5

# Package view

<i>Package Contents</i>	<i>Page</i>
<b>Interfaces</b>	
<b>ContractView</b> .....	25
Interface der Klasse View.	
<b>Classes</b>	
<b>CustomButton</b> .....	29
Ein spezieller JButton, der das Feld speichern kann zu dem er gehoert.	
<b>NameDialog</b> .....	51
Das Dialogfenster, in dem die Spieler ihre Namen eingeben muessen bevor das Spiel beginnt.	
<b>NameDialog.EnterKeyListener</b> .....	53
Ein KeyListener fuer die Enter-Taste.	
<b>View</b> .....	54
Ist fuer die Darstellung des Spiels auf dem Bildschirm verantwortlich.	
<b>View.PaintingComponent</b> .....	60
Zum Zeichnen von geometrischen Formen	
<b>WinDialog</b> .....	78
Das Dialogfenster, in dem die Spieler ihre Namen eingeben muessen bevor das Spiel beginnt.	
<b>WinDialog.EnterKeyListener</b> .....	80
Ein KeyListener fuer die Enter-Taste.	

### 5.1 Interface ContractView

Interface der Klasse View.

#### 5.1.1 Declaration

```
public interface ContractView
```

### 5.1.2 All known subinterfaces

View (in 5.5, page 54)

### 5.1.3 All classes known to implement interface

View (in 5.5, page 54)

### 5.1.4 Method summary

**allInactive()** Es wird geprueft, ob alle Buttons des Spielfeldes deaktiviert sind.  
**closeWindow()** Schliessen des Frame.  
**fillAllButtons(Spieler)** Fuellen aller Felder mit der Farbe des aktuellen Spielers.  
**füelleKasten(int, Color)** Alle Buttons in einem gewonnenen Kasten werden mit der Farbe des Spielers gefuellt, der ihn gewonnen hat.  
**setPlayerLabel(Spieler)** Der Text des PlayerLabels wird auf den Namen des aktuellen Spieler gesetzt.  
**setRepeat(boolean)** Setzt den Wunsch auf ein neues Spiel.  
**setWuerfelLabel(int)** Der Text des WuerfelLabels wird auf die aktuell gewuerfelte Zahl gesetzt.  
**shouldRepeat()** Gibt zurueck ob ein neues Spiel gewuenscht ist.  
**show(Kasten[], Spieler[], ActionListener, ActionListener)** Initialisiert das Hauptfenster und zeigt den Dialog fuer die Namenseingabe an.  
**updateButtons()** Das Aussehen der Buttons wird aktualisiert.  
**updatePoints(Spieler[])** Der aktuelle Punktestand der Spieler wird mit der View synchronisiert.

### 5.1.5 Methods

- **allInactive**

**boolean allInactive()**

- **Description**

Es wird geprueft, ob alle Buttons des Spielfeldes deaktiviert sind.

- **Returns** – Gibt true zurueck, wenn alle Buttons inaktiv sind.

- **closeWindow**

**void closeWindow()**

- **Description**

Schliessen des Frame.

- **fillAllButtons**

```
void fillAllButtons(data.Spieler spieler)
```

- **Description**

Fuellen aller Felder mit der Farbe des aktuellen Spielers. Wird bei einem Gewinn benoetigt.

- **Parameters**

- \* `spieler` – Aktueller Spieler.

- **fuelleKasten**

```
void fuelleKasten(int kastenIndex, java.awt.Color color)
```

- **Description**

Alle Buttons in einem gewonnenen Kasten werden mit der Farbe des Spielers gefuellt, der ihn gewonnen hat.

- **Parameters**

- \* `kastenIndex` – - Der Index des gewonnenen Kastens.

- \* `color` – - Die Farbe des Spieler.

- **setPlayerLabel**

```
void setPlayerLabel(data.Spieler spieler)
```

- **Description**

Der Text des PlayerLabels wird auf den Namen des aktuellen Spieler gesetzt.

- **Parameters**

- \* `spieler` – - Der aktuelle Spieler.

- **setRepeat**

```
void setRepeat(boolean repeat)
```

- **Description**

Setzt den Wunsch auf ein neues Spiel.

- **Parameters**

- \* `repeat` – true wenn ein neues Spiel gewuenscht ist. Sonst false.

- **setWuerfelLabel**

```
void setWuerfelLabel(int wuerfelErgebnis)
```

- **Description**

Der Text des WuerfelLabels wird auf die aktuell gewuerfelte Zahl gesetzt.

- **Parameters**

\* `wuerfelErgebnis` – - Das aktuelle wuerfelErgebnis.

- **shouldRepeat**

**boolean** `shouldRepeat()`

- **Description**

Gibt zurueck ob ein neues Spiel gewuenscht ist.

- **Returns** – true wenn ein neues Spiel gewuenscht ist. Sonst false.

- **show**

**void** `show(data.Kasten[] kaesten, data.Spieler[] spieler, java.awt.event.ActionListener buttonListener, java.awt.event.ActionListener restartListener)`

- **Description**

Initialisiert das Hauptfenster und zeigt den Dialog fuer die Namenseingabe an.

- **Parameters**

\* `kaesten` – Das Spielfeld.

\* `spieler` – Die Spieler.

\* `buttonListener` – Der ActionListener fuer die Buttons des Spielfeldes.

\* `restartListener` – Der ActionListener fuer den Neustart-Button.

- **updateButtons**

**void** `updateButtons()`

- **Description**

Das Aussehen der Buttons wird aktualisiert. Nicht moegliche Felder werden deaktiviert, mgliche Felder werden hervorgehoben.

- **updatePoints**

**void** `updatePoints(data.Spieler[] spieler)`

- **Description**

Der aktuelle Punktestand der Spieler wird mit der View synchronisiert.

- **Parameters**

\* `spieler` – Die beiden Spieler

## 5.2 Class CustomButton

Ein spezieller JButton, der das Feld speichern kann zu dem er gehoert.

### 5.2.1 Declaration

```
public class CustomButton
    extends javax.swing.JButton
```

### 5.2.2 Field summary

**defaultBorder** Speichert die Standardborder.  
**defaultColor** Speichert die Standardfarbe.  
**feld** Das Feld zu dem der Button gehoert.  
**kastenIndex** Index des Kasten zu dem der Button gehoert.  
**serialVersionUID** UID, welche Java haben moechte.

### 5.2.3 Constructor summary

**CustomButton(Feld, int)** Ein Spezialkonstruktor, dem anstatt dem Text ein Feld uebergeben wird.  
**CustomButton(String)** Der Standardkonstruktor eines Buttons.

### 5.2.4 Method summary

**getFeld()** Gibt das Feld des Button zurueck.  
**getKastenIndex()** Gibt den Index des Kasten zurueck in dem das Feld des Button ist.  
**highlightButton()** Es wird geprueft, ob der Button hervorgehoben werden soll und wenn ja in welcher Form.  
**setDefaultBackground()** Die Standardhintergrundfarbe wird wiederhergestellt.  
**setDeleteHighlight()** Aendert den Rahmen des Buttons, um ihn zum loeschen hervorzuheben.  
**setHighlight()** Aendert den Rahmen des Buttons, um ihn zum Setzen hervorzuheben.

### 5.2.5 Fields

- **private static final long serialVersionUID**
  - UID, welche Java haben moechte.
- **private data.Feld feld**
  - Das Feld zu dem der Button gehoert.
- **private int kastenIndex**
  - Index des Kasten zu dem der Button gehoert.
- **private final javax.swing.border.Border defaultBorder**



- Speichert die Standardborder.
- `private final java.awt.Color defaultColor`
  - Speichert die Standardfarbe.

### 5.2.6 Constructors

- **CustomButton**

```
public CustomButton(data.Feld feld, int kastenIndex)
```

- **Description**  
Ein Spezialkonstruktor, dem anstatt dem Text ein Feld uebergeben wird.
- **Parameters**
  - \* `feld` – Das Feld, zu dem der Button gehoeren soll.
  - \* `kastenIndex` – Der Index des Kastens in dem sich das Feld befindet.

- **CustomButton**

```
public CustomButton(java.lang.String s)
```

- **Description**  
Der Standardkonstruktor eines Buttons.
- **Parameters**
  - \* `s` – Der Text, den der Button haben soll.

### 5.2.7 Methods

- **getFeld**

```
public data.Feld getFeld()
```

- **Description**  
Gibt das Feld des Button zurueck.
- **Returns** – Das Feld des Button.

- **getKastenIndex**

```
public int getKastenIndex()
```

- **Description**  
Gibt den Index des Kasten zurueck in dem das Feld des Button ist.

- **Returns** – Des Kastenindex zum Button.

- **highlightButton**

```
public void highlightButton()
```

- **Description**

Es wird geprueft, ob der Button hervorgehoben werden soll und wenn ja in welcher Form.

- **setDefaultBackground**

```
public void setDefaultBackground()
```

- **Description**

Die Standardhintergrundfarbe wird wiederhergestellt.

- **setDeleteHighlight**

```
private void setDeleteHighlight()
```

- **Description**

Aendert den Rahmen des Buttons, um ihn zum loeschen hervorzuheben.

- **setHighlight**

```
private void setHighlight()
```

- **Description**

Aendert den Rahmen des Buttons, um ihn zum Setzen hervorzuheben.

### 5.2.8 Members inherited from class JButton

```
javax.swing.JButton
```

- `public AccessibleContext getAccessibleContext()`
- `public String getUIClassID()`
- `public boolean isDefaultButton()`
- `public boolean isDefaultCapable()`
- `protected String paramString()`
- `public void removeNotify()`
- `public void setDefaultCapable(boolean arg0)`
- `private static final uiClassID`
- `public void updateUI()`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`

### 5.2.9 Members inherited from class AbstractButton

javax.swing.AbstractButton

- private action
- protected actionListener
- protected void actionPerformed(Action arg0, java.lang.String arg1)
- private actionPerformedChangeListener
- public void addActionListener(java.awt.event.ActionListener arg0)
- public void addChangeListener(event.ChangeListener arg0)
- protected void addImpl(java.awt.Component arg0, java.lang.Object arg1, int arg2)
- public void addItemListener(java.awt.event.ItemListener arg0)
- public static final BORDER\_PAINTED\_CHANGED\_PROPERTY
- private borderPaintedSet
- protected transient changeEvent
- protected changeListener
- protected int checkHorizontalKey(int arg0, java.lang.String arg1)
- protected int checkVerticalKey(int arg0, java.lang.String arg1)
- void clientPropertyChange(java.lang.Object arg0, java.lang.Object arg1, java.lang.Object arg2)
- protected void configurePropertiesFromAction(Action arg0)
- public static final CONTENT\_AREA\_FILLED\_CHANGED\_PROPERTY
- private contentAreaFilled
- private contentAreaFilledSet
- protected ActionListener createActionListener()
- protected PropertyChangeListener createActionPropertyChangeListener(Action arg0)
- PropertyChangeListener createActionPropertyChangeListener0(Action arg0)
- protected ChangeListener createChangeListener()
- protected ItemListener createItemListener()
- defaultCapable
- private defaultIcon
- private defaultMargin
- public static final DISABLED\_ICON\_CHANGED\_PROPERTY
- public static final DISABLED\_SELECTED\_ICON\_CHANGED\_PROPERTY
- private disabledIcon
- private disabledSelectedIcon
- public void doClick()
- public void doClick(int arg0)
- protected void fireActionPerformed(java.awt.event.ActionEvent arg0)
- protected void fireItemStateChanged(java.awt.event.ItemEvent arg0)
- protected void fireStateChanged()
- public static final FOCUS\_PAINTED\_CHANGED\_PROPERTY
- public Action getAction()
- public String getActionCommand()
- public ActionListener getActionListeners()
- public ChangeListener getChangeListeners()
- public Icon getDisabledIcon()
- public Icon getDisabledSelectedIcon()
- public int getDisplayedMnemonicIndex()
- private AbstractButton.Handler getHandler()
- public boolean getHideActionText()
- public int getHorizontalAlignment()
- public int getHorizontalTextPosition()
- public Icon getIcon()
- public int getIconTextGap()

- `public ItemListener getItemListeners()`
- `public String getLabel()`
- `public Insets getMargin()`
- `public int getMnemonic()`
- `public ButtonModel getModel()`
- `public long getMultiClickThreshold()`
- `public Icon getPressedIcon()`
- `public Icon getRolloverIcon()`
- `public Icon getRolloverSelectedIcon()`
- `public Icon getSelectedIcon()`
- `public Object getSelectedObjects()`
- `public String getText()`
- `public ButtonUI getUI()`
- `public int getVerticalAlignment()`
- `public int getVerticalTextPosition()`
- `private handler`
- `private hideActionText`
- `public static final HORIZONTAL_ALIGNMENT_CHANGED_PROPERTY`
- `public static final HORIZONTAL_TEXT_POSITION_CHANGED_PROPERTY`
- `private horizontalAlignment`
- `private horizontalTextPosition`
- `public static final ICON_CHANGED_PROPERTY`
- `private iconTextGap`
- `private iconTextGapSet`
- `public boolean imageUpdate(java.awt.Image arg0, int arg1, int arg2, int arg3, int arg4, int arg5)`
- `protected void init(java.lang.String arg0, Icon arg1)`
- `public boolean isBorderPainted()`
- `public boolean isContentAreaFilled()`
- `public boolean isFocusPainted()`
- `private boolean isListener(java.lang.Class arg0, java.awt.event.ActionListener arg1)`
- `public boolean isRolloverEnabled()`
- `public boolean isSelected()`
- `protected itemListener`
- `void largeIconChanged(Action arg0)`
- `private margin`
- `public static final MARGIN_CHANGED_PROPERTY`
- `private mnemonic`
- `public static final MNEMONIC_CHANGED_PROPERTY`
- `private mnemonicIndex`
- `protected model`
- `public static final MODEL_CHANGED_PROPERTY`
- `private multiClickThreshold`
- `private paintBorder`
- `protected void paintBorder(java.awt.Graphics arg0)`
- `private paintFocus`
- `protected String paramString()`
- `public static final PRESSED_ICON_CHANGED_PROPERTY`
- `private pressedIcon`
- `public void removeActionListener(java.awt.event.ActionListener arg0)`
- `public void removeChangeListener(event.ChangeListener arg0)`
- `public void removeItemListener(java.awt.event.ItemListener arg0)`
- `public void removeNotify()`
- `public static final ROLLOVER_ENABLED_CHANGED_PROPERTY`
- `public static final ROLLOVER_ICON_CHANGED_PROPERTY`

- public static final **ROLLOVER\_SELECTED\_ICON\_CHANGED\_PROPERTY**
- private rolloverEnabled
- private rolloverEnabledSet
- private rolloverIcon
- private rolloverSelectedIcon
- public static final **SELECTED\_ICON\_CHANGED\_PROPERTY**
- private selectedIcon
- public void setAction(Action arg0)
- public void setActionCommand(java.lang.String arg0)
- private void setActionCommandFromAction(Action arg0)
- public void setBorderPainted(boolean arg0)
- public void setContentAreaFilled(boolean arg0)
- public void setDisabledIcon(Icon arg0)
- public void setDisabledSelectedIcon(Icon arg0)
- public void setDisplayedMnemonicIndex(int arg0) throws java.lang.IllegalArgumentException
- private void setDisplayedMnemonicIndexFromAction(Action arg0, boolean arg1)
- public void setEnabled(boolean arg0)
- public void setFocusPainted(boolean arg0)
- public void setHideActionText(boolean arg0)
- public void setHorizontalAlignment(int arg0)
- public void setHorizontalTextPosition(int arg0)
- public void setIcon(Icon arg0)
- void setIconFromAction(Action arg0)
- public void setIconTextGap(int arg0)
- public void setLabel(java.lang.String arg0)
- private setLayout
- public void setLayout(java.awt.LayoutManager arg0)
- public void setMargin(java.awt.Insets arg0)
- public void setMnemonic(char arg0)
- public void setMnemonic(int arg0)
- private void setMnemonicFromAction(Action arg0)
- public void setModel(ButtonModel arg0)
- public void setMultiClickThreshold(long arg0)
- public void setPressedIcon(Icon arg0)
- public void setRolloverEnabled(boolean arg0)
- public void setRolloverIcon(Icon arg0)
- public void setRolloverSelectedIcon(Icon arg0)
- public void setSelected(boolean arg0)
- private void setSelectedFromAction(Action arg0)
- public void setSelectedIcon(Icon arg0)
- public void setText(java.lang.String arg0)
- private void setTextFromAction(Action arg0, boolean arg1)
- public void setUI(plaf.ButtonUI arg0)
- void setUIProperty(java.lang.String arg0, java.lang.Object arg1)
- public void setVerticalAlignment(int arg0)
- public void setVerticalTextPosition(int arg0)
- boolean shouldUpdateSelectedStateFromAction()
- void smallIconChanged(Action arg0)
- private text
- public static final **TEXT\_CHANGED\_PROPERTY**
- private void updateDisplayedMnemonicIndex(java.lang.String arg0, int arg1)
- private void updateMnemonicProperties()
- public void updateUI()
- public static final **VERTICAL\_ALIGNMENT\_CHANGED\_PROPERTY**
- public static final **VERTICAL\_TEXT\_POSITION\_CHANGED\_PROPERTY**
- private verticalAlignment
- private verticalTextPosition

## 5.2.10 Members inherited from class JComponent

javax.swing.JComponent

- void **\_paintImmediately**(int arg0, int arg1, int arg2, int arg3)
- static void **()**
- private transient **aaTextInfo**
- protected **accessibleContext**
- private **actionMap**
- private static final **ACTIONMAP\_CREATED**
- public void **addAncestorListener**(event.AncestorListener arg0)
- public void **addNotify**()
- public synchronized void **addVetoableChangeListener**(java.beans.VetoableChangeListener arg0)
- private void **adjustPaintFlags**()
- private **alignmentX**
- private **alignmentY**
- boolean **alwaysOnTop**()
- private static final **ANCESTOR\_INPUTMAP\_CREATED**
- private static final **ANCESTOR\_USING\_BUFFER**
- private **ancestorInputMap**
- private **autoscrolls**
- private static final **AUTOSCROLLS\_SET**
- private **border**
- boolean **checkIfChildObscuredBySibling**()
- private transient **clientProperties**
- void **clientPropertyChanged**(java.lang.Object arg0, java.lang.Object arg1, java.lang.Object arg2)
- private static final **COMPLETELY\_OBSCURED**
- void **componentInputMapChanged**(ComponentInputMap arg0)
- private static **componentObtainingGraphicsFrom**
- private static **componentObtainingGraphicsFromLock**
- static final void **computeVisibleRect**(java.awt.Component arg0, java.awt.Rectangle arg1)
- public void **computeVisibleRect**(java.awt.Rectangle arg0)
- void **compWriteObjectNotify**()
- public boolean **contains**(int arg0, int arg1)
- private static final **CREATED\_DOUBLE\_BUFFER**
- public JToolTip **createToolTip**()
- static **DEBUG\_GRAPHICS\_LOADED**
- private static final **defaultLocale**
- private void **deregisterNextFocusableComponent**()
- public void **disable**()
- void **dndDone**()
- TransferHandler.DropLocation **dropLocationForPoint**(java.awt.Point arg0)
- public void **enable**()
- private static Rectangle **fetchRectangle**()
- public void **firePropertyChange**(java.lang.String arg0, boolean arg1, boolean arg2)
- public void **firePropertyChange**(java.lang.String arg0, char arg1, char arg2)
- public void **firePropertyChange**(java.lang.String arg0, int arg1, int arg2)
- protected void **fireVetoableChange**(java.lang.String arg0, java.lang.Object arg1, java.lang.Object arg2) throws java.beans.PropertyVetoException
- private **flags**
- private static final **FOCUS\_INPUTMAP\_CREATED**
- private static final **FOCUS\_TRAVERSAL\_KEYS\_BACKWARD\_SET**
- private static final **FOCUS\_TRAVERSAL\_KEYS\_FORWARD\_SET**

- `static final focusController`
- `private focusInputMap`
- `public AccessibleContext getAccessibleContext()`
- `public ActionListener getActionForKeyStroke(KeyStroke arg0)`
- `public final ActionMap getActionMap()`
- `final ActionMap getActionMap(boolean arg0)`
- `public float getAlignmentX()`
- `public float getAlignmentY()`
- `public AncestorListener getAncestorListeners()`
- `private AncestorNotifier getAncestorNotifier()`
- `public boolean getAutoscrolls()`
- `public int getBaseline(int arg0, int arg1)`
- `public Component.BaselineResizeBehavior getBaselineResizeBehavior()`
- `public Border getBorder()`
- `public Rectangle getBounds(java.awt.Rectangle arg0)`
- `private ArrayTable getClientProperties()`
- `public final Object getClientProperty(java.lang.Object arg0)`
- `protected Graphics getComponentGraphics(java.awt.Graphics arg0)`
- `public JPopupMenu getComponentPopupMenu()`
- `public int getConditionForKeyStroke(KeyStroke arg0)`
- `boolean getCreatedDoubleBuffer()`
- `public int getDebugGraphicsOptions()`
- `public static Locale getDefaultLocale()`
- `private boolean getFlag(int arg0)`
- `public FontMetrics getFontMetrics(java.awt.Font arg0)`
- `public Graphics getGraphics()`
- `static void getGraphicsInvoked(java.awt.Component arg0)`
- `public int getHeight()`
- `public boolean getInheritsPopupMenu()`
- `public final InputMap getInputMap()`
- `public final InputMap getInputMap(int arg0)`
- `final InputMap getInputMap(int arg0, boolean arg1)`
- `public InputVerifier getInputVerifier()`
- `public Insets getInsets()`
- `public Insets getInsets(java.awt.Insets arg0)`
- `public EventListener getListeners(java.lang.Class arg0)`
- `public Point getLocation(java.awt.Point arg0)`
- `static Set getManagingFocusBackwardTraversalKeys()`
- `static Set getManagingFocusForwardTraversalKeys()`
- `public Dimension getMaximumSize()`
- `public Dimension getMinimumSize()`
- `public Component getNextFocusableComponent()`
- `private int getObscuredState(int arg0, int arg1, int arg2, int arg3, int arg4)`
- `public Point getPopupLocation(java.awt.event.MouseEvent arg0)`
- `public Dimension getPreferredSize()`
- `public KeyStroke getRegisteredKeyStrokes()`
- `public JRootPane getRootPane()`
- `public Dimension getSize(java.awt.Dimension arg0)`
- `public Point getToolTipLocation(java.awt.event.MouseEvent arg0)`
- `public String getToolTipText()`
- `public String getToolTipText(java.awt.event.MouseEvent arg0)`
- `public Container getTopLevelAncestor()`
- `public TransferHandler getTransferHandler()`
- `public String getUIClassID()`
- `public boolean getVerifyInputWhenFocusTarget()`
- `public synchronized VetoableChangeListener getVetoableChangeListeners()`

- `public Rectangle getVisibleRect()`
- `public int getWidth()`
- `static byte getWriteObjCounter(JComponent arg0)`
- `public int getX()`
- `public int getY()`
- `public void grabFocus()`
- `private static final INHERITS_POPUP_MENU`
- `private static final INPUT_VERIFIER_SOURCE_KEY`
- `private inputVerifier`
- `private static final IS_DOUBLE_BUFFERED`
- `private static final IS_OPAQUE`
- `private static final IS_PAINTING_TILE`
- `private static final IS_PRINTING`
- `private static final IS_PRINTING_ALL`
- `private static final IS_REPAINTING`
- `private isAlignmentXSet`
- `private isAlignmentYSet`
- `private static boolean isComponentObtainingGraphicsFrom(java.awt.Component arg0)`
- `public boolean isDoubleBuffered()`
- `public static boolean isLightweightComponent(java.awt.Component arg0)`
- `public boolean isManagingFocus()`
- `public boolean isOpaque()`
- `public boolean isOptimizedDrawingEnabled()`
- `boolean isPainting()`
- `public final boolean isPaintingForPrint()`
- `protected boolean isPaintingOrigin()`
- `public boolean isPaintingTile()`
- `public boolean isRequestFocusEnabled()`
- `public boolean isValidRoot()`
- `private static final KEY_EVENTS_ENABLED`
- `private static final KEYBOARD_BINDINGS_KEY`
- `protected listenerList`
- `private static managingFocusBackwardTraversalKeys`
- `private static managingFocusForwardTraversalKeys`
- `private static final NEXT_FOCUS`
- `private static final NOT_OBSCURED`
- `private static final OPAQUE_SET`
- `public void paint(java.awt.Graphics arg0)`
- `protected void paintBorder(java.awt.Graphics arg0)`
- `protected void paintChildren(java.awt.Graphics arg0)`
- `protected void paintComponent(java.awt.Graphics arg0)`
- `void paintForceDoubleBuffered(java.awt.Graphics arg0)`
- `public void paintImmediately(int arg0, int arg1, int arg2, int arg3)`
- `public void paintImmediately(java.awt.Rectangle arg0)`
- `transient paintingChild`
- `void paintToOffscreen(java.awt.Graphics arg0, int arg1, int arg2, int arg3, int arg4, int arg5, int arg6)`
- `protected String paramString()`
- `private static final PARTIALLY_OBSCURED`
- `private popupMenu`
- `public void print(java.awt.Graphics arg0)`
- `public void printAll(java.awt.Graphics arg0)`
- `protected void printBorder(java.awt.Graphics arg0)`
- `protected void printChildren(java.awt.Graphics arg0)`
- `protected void printComponent(java.awt.Graphics arg0)`



- protected void **processComponentKeyEvent**(java.awt.event.KeyEvent arg0)
- protected boolean **processKeyBinding**(KeyStroke arg0, java.awt.event.KeyEvent arg1, int arg2, boolean arg3)
- boolean **processKeyBindings**(java.awt.event.KeyEvent arg0, boolean arg1)
- static boolean **processKeyBindingsForAllComponents**(java.awt.event.KeyEvent arg0, java.awt.Container arg1, boolean arg2)
- protected void **processKeyEvent**(java.awt.event.KeyEvent arg0)
- protected void **processMouseEvent**(java.awt.event.MouseEvent arg0)
- protected void **processMouseMotionEvent**(java.awt.event.MouseEvent arg0)
- public final void **putClientProperty**(java.lang.Object arg0, java.lang.Object arg1)
- private void **readObject**(java.io.ObjectInputStream arg0) throws java.io.IOException, java.lang.ClassNotFoundException
- private static final **readObjectCallbacks**
- boolean **rectangleIsObscured**(int arg0, int arg1, int arg2, int arg3)
- private static void **recycleRectangle**(java.awt.Rectangle arg0)
- public void **registerKeyboardAction**(java.awt.event.ActionListener arg0, KeyStroke arg1, int arg2)
- public void **registerKeyboardAction**(java.awt.event.ActionListener arg0, java.lang.String arg1, KeyStroke arg2, int arg3)
- private void **registerNextFocusableComponent**()
- private void **registerNextFocusableComponent**(java.awt.Component arg0)
- private void **registerWithKeyboardManager**(boolean arg0)
- private void **registerWithKeyboardManager**(KeyStroke arg0)
- public void **removeAncestorListener**(event.AncestorListener arg0)
- public void **removeNotify**()
- public synchronized void **removeVetoableChangeListener**(java.beans.VetoableChangeListener arg0)
- public void **repaint**(long arg0, int arg1, int arg2, int arg3, int arg4)
- public void **repaint**(java.awt.Rectangle arg0)
- private static final **REQUEST\_FOCUS\_DISABLED**
- public boolean **requestDefaultFocus**()
- public void **requestFocus**()
- public boolean **requestFocus**(boolean arg0)
- public boolean **requestFocusInWindow**()
- protected boolean **requestFocusInWindow**(boolean arg0)
- private static final **RESERVED\_1**
- private static final **RESERVED\_2**
- private static final **RESERVED\_3**
- private static final **RESERVED\_4**
- private static final **RESERVED\_5**
- private static final **RESERVED\_6**
- public void **resetKeyboardActions**()
- public void **reshape**(int arg0, int arg1, int arg2, int arg3)
- private static final **REVALIDATE\_RUNNABLE\_SCHEDULED**
- public void **revalidate**()
- static Graphics **safelyGetGraphics**(java.awt.Component arg0)
- static Graphics **safelyGetGraphics**(java.awt.Component arg0, java.awt.Component arg1)
- public void **scrollRectToVisible**(java.awt.Rectangle arg0)
- public final void **setActionMap**(ActionMap arg0)
- public void **setAlignmentX**(float arg0)
- public void **setAlignmentY**(float arg0)
- public void **setAutoscrolls**(boolean arg0)
- public void **setBackground**(java.awt.Color arg0)
- public void **setBorder**(border.Border arg0)

- `public void setComponentPopupMenu(JPopupMenu arg0)`
- `void setCreatedDoubleBuffer(boolean arg0)`
- `public void setDebugGraphicsOptions(int arg0)`
- `public static void setDefaultLocale(java.util.Locale arg0)`
- `public void setDoubleBuffered(boolean arg0)`
- `Object setDropLocation(TransferHandler.DropLocation arg0, java.lang.Object arg1, boolean arg2)`
- `public void setEnabled(boolean arg0)`
- `private void setFlag(int arg0, boolean arg1)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFont(java.awt.Font arg0)`
- `public void setForeground(java.awt.Color arg0)`
- `public void setInheritsPopupMenu(boolean arg0)`
- `public final void setInputMap(int arg0, InputMap arg1)`
- `public void setInputVerifier(InputVerifier arg0)`
- `public void setMaximumSize(java.awt.Dimension arg0)`
- `public void setMinimumSize(java.awt.Dimension arg0)`
- `public void setNextFocusableComponent(java.awt.Component arg0)`
- `public void setOpaque(boolean arg0)`
- `void setPaintingChild(java.awt.Component arg0)`
- `public void setPreferredSize(java.awt.Dimension arg0)`
- `public void setRequestFocusEnabled(boolean arg0)`
- `public void setToolTipText(java.lang.String arg0)`
- `public void setTransferHandler(TransferHandler arg0)`
- `protected void setUI(plaf.ComponentUI arg0)`
- `void setUIProperty(java.lang.String arg0, java.lang.Object arg1)`
- `public void setVerifyInputWhenFocusTarget(boolean arg0)`
- `public void setVisible(boolean arg0)`
- `static void setWriteObjCounter(JComponent arg0, byte arg1)`
- `int shouldDebugGraphics()`
- `void superProcessMouseEvent(java.awt.event.MouseEvent arg0)`
- `private static tempRectangles`
- `public static final TOOL_TIP_TEXT_KEY`
- `protected transient ui`
- `private static final uiClassID`
- `public static final UNDEFINED_CONDITION`
- `private void uninstallUIAndProperties()`
- `public void unregisterKeyboardAction(KeyStroke arg0)`
- `private void unregisterWithKeyboardManager()`
- `private void unregisterWithKeyboardManager(KeyStroke arg0)`
- `public void update(java.awt.Graphics arg0)`
- `public void updateUI()`
- `private verifyInputWhenFocusTarget`
- `private vetoableChangeSupport`
- `public static final WHEN_ANCESTOR_OF_FOCUSED_COMPONENT`
- `public static final WHEN_FOCUSED`
- `public static final WHEN_IN_FOCUSED_WINDOW`
- `private static final WHEN_IN_FOCUSED_WINDOW_BINDINGS`
- `private static final WIF_INPUTMAP_CREATED`
- `private windowInputMap`
- `private static final WRITE_OBJ_COUNTER_FIRST`
- `private static final WRITE_OBJ_COUNTER_LAST`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`

## 5.2.11 Members inherited from class Container

```
java.awt.Container
```

- static void ()
- public Component add(Component arg0)
- public Component add(Component arg0, int arg1)
- public void add(Component arg0, java.lang.Object arg1)
- public void add(Component arg0, java.lang.Object arg1, int arg2)
- public Component add(java.lang.String arg0, Component arg1)
- public synchronized void addContainerListener(event.ContainerListener arg0)
- private void addDelicately(Component arg0, Container arg1, int arg2)
- protected void addImpl(Component arg0, java.lang.Object arg1, int arg2)
- public void addNotify()
- public void addPropertyChangeListener(java.beans.PropertyChangeListener arg0)
- public void addPropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)
- void adjustDecendantsOnParent(int arg0)
- void adjustDescendants(int arg0)
- void adjustListeningChildren(long arg0, int arg1)
- public void applyComponentOrientation(ComponentOrientation arg0)
- public boolean areFocusTraversalKeysSet(int arg0)
- boolean canContainFocusOwner(Component arg0)
- private void checkAdding(Component arg0, int arg1)
- private void checkAddToSelf(Component arg0)
- void checkGD(java.lang.String arg0)
- private void checkNotAWindow(Component arg0)
- void clearCurrentFocusCycleRootOnHide()
- void clearMostRecentFocusOwnerOnHide()
- private component
- transient containerListener
- private containerSerializedDataVersion
- final boolean containsFocus()
- public int countComponents()
- int countHierarchyMembers()
- final void createChildHierarchyEvents(int arg0, long arg1, boolean arg2)
- final int createHierarchyEvents(int arg0, Component arg1, Container arg2, long arg3, boolean arg4)
- final void decreaseComponentCount(Component arg0)
- public void deliverEvent(Event arg0)
- transient descendantsCount
- private static descendUnconditionallyWhenValidating
- private dispatcher
- void dispatchEventImpl(AWTEvent arg0)
- void dispatchEventToSelf(AWTEvent arg0)
- public void doLayout()
- private static final EMPTY\_ARRAY
- boolean eventEnabled(AWTEvent arg0)
- private static final eventLog
- public Component findComponentAt(int arg0, int arg1)
- final Component findComponentAt(int arg0, int arg1, boolean arg2)
- public Component findComponentAt(Point arg0)
- final Component findComponentAtImpl(int arg0, int arg1, boolean arg2)
- private Container findTraversalRoot()
- private focusCycleRoot
- private transient focusTraversalPolicy
- private focusTraversalPolicyProvider

- Accessible getAccessibleAt(Point arg0)
- Accessible getAccessibleChild(int arg0)
- int getAccessibleChildrenCount()
- public float getAlignmentX()
- public float getAlignmentY()
- private int getBottommostComponentIndex()
- public Component getComponent(int arg0)
- public Component getComponentAt(int arg0, int arg1)
- public Component getComponentAt(Point arg0)
- public int getComponentCount()
- final Component getComponents\_NoClientCode()
- public Component getComponents()
- Component getComponentsSync()
- public int getComponentZOrder(Component arg0)
- public synchronized ContainerListener getContainerListeners()
- Component getDropTargetEventTarget(int arg0, int arg1, boolean arg2)
- public Set getFocusTraversalKeys(int arg0)
- public FocusTraversalPolicy getFocusTraversalPolicy()
- Container getHeavyweightContainer()
- public Insets getInsets()
- public LayoutManager getLayout()
- public EventListener getListeners(java.lang.Class arg0)
- private int getListenersCount(int arg0, boolean arg1)
- public Dimension getMaximumSize()
- public Dimension getMinimumSize()
- Component getMouseEventTarget(int arg0, int arg1, boolean arg2)
- private Component getMouseEventTarget(int arg0, int arg1, boolean arg2, Container.EventTargetFilter arg3, boolean arg4)
- private Component getMouseEventTargetImpl(int arg0, int arg1, boolean arg2, Container.EventTargetFilter arg3, boolean arg4, boolean arg5)
- public Point getMousePosition(boolean arg0) throws HeadlessException
- final Region getOpaqueShape()
- public Dimension getPreferredSize()
- private int getTopmostComponentIndex()
- final Container getTraversalRoot()
- final boolean hasHeavyweightDescendants()
- final boolean hasLightweightDescendants()
- static final INCLUDE\_SELF
- final void increaseComponentCount(Component arg0)
- void initializeFocusTraversalKeys()
- private static native void initIDs()
- public Insets insets()
- public void invalidate()
- void invalidateParent()
- void invalidateTree()
- public boolean isAncestorOf(Component arg0)
- public boolean isFocusCycleRoot()
- public boolean isFocusCycleRoot(Container arg0)
- public final boolean isFocusTraversalPolicyProvider()
- public boolean isFocusTraversalPolicySet()
- private static final isJavaAwtSmartInvalidate
- private boolean isParentOf(Component arg0)
- final boolean isRecursivelyVisibleUpToHeavyweightContainer()
- private static boolean isRemoveNotifyNeeded(Component arg0, Container arg1, Container arg2)
- boolean isSameOrAncestorOf(Component arg0, boolean arg1)

- `public boolean isValidRoot()`
- `public void layout()`
- `layoutMgr`
- `void lightweightPaint(Graphics arg0)`
- `void lightweightPrint(Graphics arg0)`
- `public void list(java.io.PrintStream arg0, int arg1)`
- `public void list(java.io.PrintWriter arg0, int arg1)`
- `transient listeningBoundsChildren`
- `transient listeningChildren`
- `public Component locate(int arg0, int arg1)`
- `private static final log`
- `public Dimension minimumSize()`
- `private static final mixingLog`
- `void mixOnHiding(boolean arg0)`
- `void mixOnReshaping()`
- `void mixOnShowing()`
- `void mixOnValidating()`
- `void mixOnZOrderChanging(int arg0, int arg1)`
- `transient modalAppContext`
- `transient modalComp`
- `int numListening(long arg0)`
- `private transient numOfHWComponents`
- `private transient numOfLWComponents`
- `public void paint(Graphics arg0)`
- `public void paintComponents(Graphics arg0)`
- `void paintHeavyweightComponents(Graphics arg0)`
- `protected String paramString()`
- `void postProcessKeyEvent(event.KeyEvent arg0)`
- `boolean postsOldMouseEvents()`
- `public Dimension preferredSize()`
- `void preProcessKeyEvent(event.KeyEvent arg0)`
- `transient preserveBackgroundColor`
- `public void print(Graphics arg0)`
- `public void printComponents(Graphics arg0)`
- `void printHeavyweightComponents(Graphics arg0)`
- `private transient printing`
- `private transient printingThreads`
- `protected void processContainerEvent(event.ContainerEvent arg0)`
- `protected void processEvent(AWTEvent arg0)`
- `void proxyEnableEvents(long arg0)`
- `private void readObject(java.io.ObjectInputStream arg0) throws java.lang.ClassNotFoundException, java.io.IOException`
- `final void recursiveApplyCurrentShape()`
- `final void recursiveApplyCurrentShape(int arg0)`
- `final void recursiveApplyCurrentShape(int arg0, int arg1)`
- `private void recursiveHideHeavyweightChildren()`
- `private void recursiveRelocateHeavyweightChildren(Point arg0)`
- `private void recursiveShowHeavyweightChildren()`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0)`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0, int arg1)`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0, int arg1, int arg2)`
- `public void remove(Component arg0)`
- `public void remove(int arg0)`
- `public void removeAll()`

- `public synchronized void removeContainerListener(event.ContainerListener arg0)`
- `private boolean removeDelicately(Component arg0, Container arg1, int arg2)`
- `public void removeNotify()`
- `private void reparentChild(Component arg0)`
- `private void reparentTraverse(peer.ContainerPeer arg0, Container arg1)`
- `static final SEARCH_HEAVYWEIGHTS`
- `private static final serialPersistentFields`
- `private static final serialVersionUID`
- `public void setComponentZOrder(Component arg0, int arg1)`
- `public void setFocusCycleRoot(boolean arg0)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalPolicy(FocusTraversalPolicy arg0)`
- `public final void setFocusTraversalPolicyProvider(boolean arg0)`
- `public void setFont(Font arg0)`
- `public void setLayout(LayoutManager arg0)`
- `private void startLWModal()`
- `private void stopLWModal()`
- `public void transferFocusDownCycle()`
- `public void update(Graphics arg0)`
- `boolean updateGraphicsData(GraphicsConfiguration arg0)`
- `public void validate()`
- `protected void validateTree()`
- `final void validateUnconditionally()`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`

### 5.2.12 Members inherited from class Component

`java.awt.Component`

- `static void ()`
- `private transient volatile acc`
- `accessibleContext`
- `public boolean action(Event arg0, java.lang.Object arg1)`
- `static final ActionListenerK`
- `public void add(PopupMenu arg0)`
- `public synchronized void addComponentListener(event.ComponentListener arg0)`
- `public synchronized void addFocusListener(event.FocusListener arg0)`
- `public void addHierarchyBoundsListener(event.HierarchyBoundsListener arg0)`
- `public void addHierarchyListener(event.HierarchyListener arg0)`
- `public synchronized void addInputMethodListener(event.InputMethodListener arg0)`
- `public synchronized void addKeyListener(event.KeyListener arg0)`
- `public synchronized void addMouseListener(event.MouseListener arg0)`
- `public synchronized void addMouseMotionListener(event.MouseMotionListener arg0)`
- `public synchronized void addMouseWheelListener(event.MouseWheelListener arg0)`
- `public void addNotify()`
- `public void addPropertyChangeListener(java.beans.PropertyChangeListener arg0)`
- `public void addPropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)`
- `void adjustListeningChildrenOnParent(long arg0, int arg1)`
- `static final adjustmentListenerK`
- `transient appContext`
- `public void applyComponentOrientation(ComponentOrientation arg0)`
- `void applyCompoundShape(sun.java2d.pipe.Region arg0)`

- `void applyCurrentShape()`
- `private final void applyCurrentShapeBelowMe()`
- `final boolean areBoundsValid()`
- `public boolean areFocusTraversalKeysSet(int arg0)`
- `boolean areInputMethodsEnabled()`
- `private autoFocusTransferOnDisposal`
- `void autoProcessMouseWheel(event.MouseWheelEvent arg0)`
- `background`
- `transient backgroundEraseDisabled`
- `public static final BOTTOM_ALIGNMENT`
- `public Rectangle bounds()`
- `private boundsOp`
- `transient bufferStrategy`
- `private Region calculateCurrentShape()`
- `final boolean canBeFocusOwner()`
- `final boolean canBeFocusOwnerRecursively()`
- `public static final CENTER_ALIGNMENT`
- `private changeSupport`
- `private boolean checkCoalescing()`
- `void checkGD(java.lang.String arg0)`
- `public int checkImage(Image arg0, image.ImageObserver arg1)`
- `public int checkImage(Image arg0, int arg1, int arg2, image.ImageObserver arg3)`
- `final void checkTreeLock()`
- `boolean checkWindowClosingException()`
- `void clearCurrentFocusCycleRootOnHide()`
- `void clearMostRecentFocusOwnerOnHide()`
- `protected AWTEvent coalesceEvents(AWTEvent arg0, AWTEvent arg1)`
- `private static final coalesceEventsParams`
- `private static final coalesceMap`
- `private transient coalescingEnabled`
- `transient componentListener`
- `static final componentListenerK`
- `transient componentOrientation`
- `private componentSerializedDataVersion`
- `private transient compoundShape`
- `String constructComponentName()`
- `static final containerListenerK`
- `public boolean contains(int arg0, int arg1)`
- `public boolean contains(Point arg0)`
- `boolean containsFocus()`
- `int countHierarchyMembers()`
- `void createBufferStrategy(int arg0)`
- `void createBufferStrategy(int arg0, BufferCapabilities arg1) throws AWTException`
- `int createHierarchyEvents(int arg0, Component arg1, Container arg2, long arg3, boolean arg4)`
- `public Image createImage(image.ImageProducer arg0)`
- `public Image createImage(int arg0, int arg1)`
- `public VolatileImage createVolatileImage(int arg0, int arg1)`
- `public VolatileImage createVolatileImage(int arg0, int arg1, ImageCapabilities arg2) throws AWTException`
- `cursor`
- `public void deliverEvent(Event arg0)`
- `public void disable()`
- `protected final void disableEvents(long arg0)`
- `public final void dispatchEvent(AWTEvent arg0)`

- `void dispatchEventImpl(AWTEvent arg0)`
- `boolean dispatchMouseWheelToAncestor(event.MouseWheelEvent arg0)`
- `public void doLayout()`
- `private void doSwingSerialization()`
- `dropTarget`
- `public void enable()`
- `public void enable(boolean arg0)`
- `enabled`
- `protected final void enableEvents(long arg0)`
- `public void enableInputMethods(boolean arg0)`
- `transient eventCache`
- `boolean eventEnabled(AWTEvent arg0)`
- `private static final eventLog`
- `eventMask`
- `boolean eventTypeEnabled(int arg0)`
- `Component findUnderMouseInWindow(PointerInfo arg0)`
- `protected void firePropertyChange(java.lang.String arg0, boolean arg1, boolean arg2)`
- `public void firePropertyChange(java.lang.String arg0, byte arg1, byte arg2)`
- `public void firePropertyChange(java.lang.String arg0, char arg1, char arg2)`
- `public void firePropertyChange(java.lang.String arg0, double arg1, double arg2)`
- `public void firePropertyChange(java.lang.String arg0, float arg1, float arg2)`
- `protected void firePropertyChange(java.lang.String arg0, int arg1, int arg2)`
- `public void firePropertyChange(java.lang.String arg0, long arg1, long arg2)`
- `protected void firePropertyChange(java.lang.String arg0, java.lang.Object arg1, java.lang.Object arg2)`
- `public void firePropertyChange(java.lang.String arg0, short arg1, short arg2)`
- `private static final FOCUS_TRAVERSABLE_DEFAULT`
- `private static final FOCUS_TRAVERSABLE_SET`
- `private static final FOCUS_TRAVERSABLE_UNKNOWN`
- `private focusable`
- `transient focusListener`
- `static final focusListenerK`
- `private static final focusLog`
- `private static final focusTraversalKeyPropertyNames`
- `focusTraversalKeys`
- `private focusTraversalKeysEnabled`
- `font`
- `foreground`
- `final AccessControlContext getAccessControlContext()`
- `public AccessibleContext getAccessibleContext()`
- `int getAccessibleIndexInParent()`
- `AccessibleStateSet getAccessibleStateSet()`
- `public float getAlignmentX()`
- `public float getAlignmentY()`
- `private Region getAppliedShape()`
- `Image getBackBuffer()`
- `public Color getBackground()`
- `public int getBaseline(int arg0, int arg1)`
- `public Component.BaselineResizeBehavior getBaselineResizeBehavior()`
- `public Rectangle getBounds()`
- `public Rectangle getBounds(Rectangle arg0)`
- `int getBoundsOp()`
- `BufferStrategy getBufferStrategy()`
- `public ColorModel getColorModel()`
- `public Component getComponentAt(int arg0, int arg1)`



- `public Component getComponentAt(Point arg0)`
- `public synchronized ComponentListener getComponentListeners()`
- `public ComponentOrientation getComponentOrientation()`
- `Container getContainer()`
- `Window getContainingWindow()`
- `final Cursor getCursor_NoClientCode()`
- `public Cursor getCursor()`
- `public synchronized DropTarget getDropTarget()`
- `public Container getFocusCycleRootAncestor()`
- `public synchronized FocusListener getFocusListeners()`
- `final Set getFocusTraversalKeys_NoIDCheck(int arg0)`
- `public Set getFocusTraversalKeys(int arg0)`
- `public boolean getFocusTraversalKeysEnabled()`
- `final Font getFont_NoClientCode()`
- `public Font getFont()`
- `public FontMetrics getFontMetrics(Font arg0)`
- `public Color getForeground()`
- `final Graphics getGraphics_NoClientCode()`
- `public Graphics getGraphics()`
- `final GraphicsConfiguration getGraphicsConfiguration_NoClientCode()`
- `public GraphicsConfiguration getGraphicsConfiguration()`
- `public int getHeight()`
- `public synchronized HierarchyBoundsListener getHierarchyBoundsListeners()`
- `public synchronized HierarchyListener getHierarchyListeners()`
- `final ComponentPeer getHWPeeAboveMe()`
- `public boolean getIgnoreRepaint()`
- `public InputContext getInputContext()`
- `public synchronized InputMethodListener getInputMethodListeners()`
- `public InputMethodRequests getInputMethodRequests()`
- `private Insets getInsets_NoClientCode()`
- `public synchronized KeyListener getKeyListeners()`
- `public EventListener getListeners(java.lang.Class arg0)`
- `public Locale getLocale()`
- `public Point getLocation()`
- `public Point getLocation(Point arg0)`
- `final Point getLocationOnScreen_NoTreeLock()`
- `public Point getLocationOnScreen()`
- `Point getLocationOnWindow()`
- `public Dimension getMaximumSize()`
- `public Dimension getMinimumSize()`
- `public synchronized MouseListener getMouseListeners()`
- `public synchronized MouseMotionListener getMouseMotionListeners()`
- `public Point getMousePosition() throws HeadlessException`
- `public synchronized MouseWheelListener getMouseWheelListeners()`
- `public String getName()`
- `Container getNativeContainer()`
- `final Component getNextFocusCandidate()`
- `final Region getNormalShape()`
- `Object getObjectLock()`
- `Region getOpaqueShape()`
- `final Container getParent_NoClientCode()`
- `public Container getParent()`
- `public ComponentPeer getPeer()`
- `public Dimension getPreferredSize()`
- `public PropertyChangeListener getPropertyChangeListeners()`
- `public PropertyChangeListener getPropertyChangeListeners(java.lang.String arg0)`

- final int `getSiblingIndexAbove()`
- final int `getSiblingIndexBelow()`
- public Dimension `getSize()`
- public Dimension `getSize(Dimension arg0)`
- public Toolkit `getToolkit()`
- final Toolkit `getToolkitImpl()`
- Container `getTraversalRoot()`
- public final Object `getTreeLock()`
- public int `getWidth()`
- public int `getX()`
- public int `getY()`
- public boolean `gotFocus(Event arg0, java.lang.Object arg1)`
- private transient `graphicsConfig`
- public boolean `handleEvent(Event arg0)`
- public boolean `hasFocus()`
- height
- public void `hide()`
- transient `hierarchyBoundsListener`
- static final `hierarchyBoundsListenerK`
- transient `hierarchyListener`
- static final `hierarchyListenerK`
- ignoreRepaint
- public boolean `imageUpdate(Image arg0, int arg1, int arg2, int arg3, int arg4, int arg5)`
- static `incRate`
- void `initializeFocusTraversalKeys()`
- private static native void `initIDs()`
- transient `inputMethodListener`
- static final `inputMethodListenerK`
- public boolean `inside(int arg0, int arg1)`
- public void `invalidate()`
- final void `invalidateIfValid()`
- void `invalidateParent()`
- private transient `isAddNotifyComplete`
- boolean `isAutoFocusTransferOnDisposal()`
- public boolean `isBackgroundSet()`
- private static boolean `isCoalesceEventsOverridden(java.lang.Class arg0)`
- final boolean `isCoalescingEnabled()`
- public boolean `isCursorSet()`
- public boolean `isDisplayable()`
- public boolean `isDoubleBuffered()`
- public boolean `isEnabled()`
- final boolean `isEnabledImpl()`
- public boolean `isFocusable()`
- public boolean `isFocusCycleRoot(Container arg0)`
- public boolean `isFocusOwner()`
- public boolean `isFocusTraversable()`
- private `isFocusTraversableOverridden`
- final boolean `isFocusTraversableOverridden()`
- public boolean `isFontSet()`
- public boolean `isForegroundSet()`
- static `isInc`
- static boolean `isInstanceOf(java.lang.Object arg0, java.lang.String arg1)`
- public boolean `isLightweight()`
- public boolean `isMaximumSizeSet()`
- public boolean `isMinimumSizeSet()`

- final boolean isMixingNeeded()
- final boolean isNonOpaqueForMixing()
- public boolean isOpaque()
- isPacked
- public boolean isPreferredSizeSet()
- boolean isRecursivelyVisible()
- private boolean isRequestFocusAccepted(boolean arg0, boolean arg1, sun.awt.CausedFocusEvent.Cause arg2)
- boolean isSameOrAncestorOf(Component arg0, boolean arg1)
- public boolean isShowing()
- public boolean isValid()
- final boolean isVisible\_NoClientCode()
- public boolean isVisible()
- static final itemListenerK
- public boolean keyDown(Event arg0, int arg1)
- transient keyListener
- static final keyListenerK
- public boolean keyUp(Event arg0, int arg1)
- public void layout()
- public static final LEFT\_ALIGNMENT
- void lightweightPaint(Graphics arg0)
- void lightweightPrint(Graphics arg0)
- public void list()
- public void list(java.io.PrintStream arg0)
- public void list(java.io.PrintStream arg0, int arg1)
- public void list(java.io.PrintWriter arg0)
- public void list(java.io.PrintWriter arg0, int arg1)
- locale
- public Component locate(int arg0, int arg1)
- private Point location\_NoClientCode()
- public Point location()
- static final LOCK
- private static final log
- public boolean lostFocus(Event arg0, java.lang.Object arg1)
- maxSize
- maxSizeSet
- public Dimension minimumSize()
- minSize
- minSizeSet
- private transient mixingCutoutRegion
- private static final mixingLog
- void mixOnHiding(boolean arg0)
- void mixOnReshaping()
- void mixOnShowing()
- void mixOnValidating()
- void mixOnZOrderChanging(int arg0, int arg1)
- public boolean mouseDown(Event arg0, int arg1, int arg2)
- public boolean mouseDrag(Event arg0, int arg1, int arg2)
- public boolean mouseEnter(Event arg0, int arg1, int arg2)
- public boolean mouseExit(Event arg0, int arg1, int arg2)
- transient mouseListener
- static final mouseListenerK
- transient mouseMotionListener
- static final mouseMotionListenerK
- public boolean mouseMove(Event arg0, int arg1, int arg2)
- public boolean mouseUp(Event arg0, int arg1, int arg2)

- transient `mouseWheelListener`
- static final `mouseWheelListenerK`
- public void `move(int arg0, int arg1)`
- private `name`
- private `nameExplicitlySet`
- `newEventsOnly`
- public void `nextFocus()`
- private void `notifyNewBounds(boolean arg0, boolean arg1)`
- int `numListening(long arg0)`
- private transient `objectLock`
- static final `ownedWindowK`
- public void `paint(Graphics arg0)`
- public void `paintAll(Graphics arg0)`
- void `paintHeavyweightComponents(Graphics arg0)`
- protected String `paramString()`
- transient `parent`
- transient `peer`
- `peerFont`
- Point `pointRelativeToComponent(Point arg0)`
- `popups`
- public boolean `postEvent(Event arg0)`
- boolean `postsOldMouseEvents()`
- public Dimension `preferredSize()`
- `prefSize`
- `prefSizeSet`
- public boolean `prepareImage(Image arg0, image.ImageObserver arg1)`
- public boolean `prepareImage(Image arg0, int arg1, int arg2, image.ImageObserver arg3)`
- public void `print(Graphics arg0)`
- public void `printAll(Graphics arg0)`
- void `printHeavyweightComponents(Graphics arg0)`
- protected void `processComponentEvent(event.ComponentEvent arg0)`
- protected void `processEvent(AWTEvent arg0)`
- protected void `processFocusEvent(event.FocusEvent arg0)`
- protected void `processHierarchyBoundsEvent(event.HierarchyEvent arg0)`
- protected void `processHierarchyEvent(event.HierarchyEvent arg0)`
- protected void `processInputMethodEvent(event.InputMethodEvent arg0)`
- protected void `processKeyEvent(event.KeyEvent arg0)`
- protected void `processMouseEvent(event.MouseEvent arg0)`
- protected void `processMouseMotionEvent(event.MouseEvent arg0)`
- protected void `processMouseWheelEvent(event.MouseWheelEvent arg0)`
- private void `readObject(java.io.ObjectInputStream arg0) throws java.lang.ClassNotFoundException, java.io.IOException`
- final void `relocateComponent()`
- public void `remove(MenuComponent arg0)`
- public synchronized void `removeComponentListener(event.ComponentListener arg0)`
- public synchronized void `removeFocusListener(event.FocusListener arg0)`
- public void `removeHierarchyBoundsListener(event.HierarchyBoundsListener arg0)`
- public void `removeHierarchyListener(event.HierarchyListener arg0)`
- public synchronized void `removeInputMethodListener(event.InputMethodListener arg0)`
- public synchronized void `removeKeyListener(event.KeyListener arg0)`
- public synchronized void `removeMouseListener(event.MouseListener arg0)`
- public synchronized void `removeMouseMotionListener(event.MouseMotionListener arg0)`

- `public synchronized void removeMouseListener(event.MouseWheelListener arg0)`
- `public void removeNotify()`
- `public void removePropertyChangeListener(java.beans.PropertyChangeListener arg0)`
- `public void removePropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)`
- `public void repaint()`
- `public void repaint(int arg0, int arg1, int arg2, int arg3)`
- `public void repaint(long arg0)`
- `public void repaint(long arg0, int arg1, int arg2, int arg3, int arg4)`
- `private void repaintParentIfNeeded(int arg0, int arg1, int arg2, int arg3)`
- `public void requestFocus()`
- `protected boolean requestFocus(boolean arg0)`
- `boolean requestFocus(boolean arg0, sun.awt.CausedFocusEvent.Cause arg1)`
- `boolean requestFocus(sun.awt.CausedFocusEvent.Cause arg0)`
- `private static requestFocusController`
- `final boolean requestFocusHelper(boolean arg0, boolean arg1)`
- `final boolean requestFocusHelper(boolean arg0, boolean arg1, sun.awt.CausedFocusEvent.Cause arg2)`
- `public boolean requestFocusInWindow()`
- `protected boolean requestFocusInWindow(boolean arg0)`
- `boolean requestFocusInWindow(boolean arg0, sun.awt.CausedFocusEvent.Cause arg1)`
- `boolean requestFocusInWindow(sun.awt.CausedFocusEvent.Cause arg0)`
- `public void reshape(int arg0, int arg1, int arg2, int arg3)`
- `private void reshapeNativePeer(int arg0, int arg1, int arg2, int arg3, int arg4)`
- `public void resize(Dimension arg0)`
- `public void resize(int arg0, int arg1)`
- `public void revalidate()`
- `public static final RIGHT_ALIGNMENT`
- `private static final serialVersionUID`
- `void setAutoFocusTransferOnDisposal(boolean arg0)`
- `public void setBackground(Color arg0)`
- `public void setBounds(int arg0, int arg1, int arg2, int arg3)`
- `public void setBounds(Rectangle arg0)`
- `void setBoundsOp(int arg0)`
- `public void setComponentOrientation(ComponentOrientation arg0)`
- `public void setCursor(Cursor arg0)`
- `public synchronized void setDropTarget(dnd.DropTarget arg0)`
- `public void setEnabled(boolean arg0)`
- `public void setFocusable(boolean arg0)`
- `final void setFocusTraversalKeys_NoIDCheck(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalKeysEnabled(boolean arg0)`
- `public void setFont(Font arg0)`
- `public void setForeground(Color arg0)`
- `void setGraphicsConfiguration(GraphicsConfiguration arg0)`
- `public void setIgnoreRepaint(boolean arg0)`
- `public void setLocale(java.util.Locale arg0)`
- `public void setLocation(int arg0, int arg1)`
- `public void setLocation(Point arg0)`
- `public void setMaximumSize(Dimension arg0)`
- `public void setMinimumSize(Dimension arg0)`
- `public void setName(java.lang.String arg0)`

- `public void setPreferredSize(Dimension arg0)`
- `static synchronized void setRequestFocusController(sun.awt.RequestFocusController arg0)`
- `public void setSize(Dimension arg0)`
- `public void setSize(int arg0, int arg1)`
- `public void setVisible(boolean arg0)`
- `public void show()`
- `public void show(boolean arg0)`
- `public Dimension size()`
- `final void subtractAndApplyShape(sun.java2d.pipe.Region arg0)`
- `final void subtractAndApplyShapeBelowMe()`
- `static final textListenerK`
- `public static final TOP_ALIGNMENT`
- `public String toString()`
- `public void transferFocus()`
- `boolean transferFocus(boolean arg0)`
- `public void transferFocusBackward()`
- `boolean transferFocusBackward(boolean arg0)`
- `public void transferFocusUpCycle()`
- `public void update(Graphics arg0)`
- `final void updateCursorImmediately()`
- `boolean updateGraphicsData(GraphicsConfiguration arg0)`
- `void updateZOrder()`
- `private volatile valid`
- `public void validate()`
- `visible`
- `width`
- `transient windowClosingException`
- `static final windowFocusListenerK`
- `static final windowListenerK`
- `static final windowStateListenerK`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`
- `x`
- `y`

### 5.3 Class NameDialog

Das Dialogfenster, in dem die Spieler ihre Namen eingeben muessen bevor das Spiel beginnt.

#### 5.3.1 Declaration

```
public class NameDialog
    extends java.lang.Object
```

#### 5.3.2 Field summary

**btnAbort** Beenden Button im Dialog.

**btnOK** OK Button im Dialog.

**dialog** Dialog.

**spieler** Array mit Spielern.

**txtName1** Textfeld fuer Name.

**txtName2** Textfeld fuer Name.

### 5.3.3 Constructor summary

**NameDialog(Frame, Spieler[])** Hiermit wird das DialogFenster erstellt.

### 5.3.4 Method summary

**cropText(JTextField, int)** Kuerzt Text in einem Textfeld auf eine maximale Laenge.

**showNameDialog(Frame)** Zeigt den Namensdialog an.

**verifyNames()** Diese Methode ueberprueft die Eingabe in den beiden Textfeldern.

### 5.3.5 Fields

- **private data.Spieler[] spieler**
  - Array mit Spielern.
- **private javax.swing.JTextField txtName1**
  - Textfeld fuer Name.
- **private javax.swing.JTextField txtName2**
  - Textfeld fuer Name.
- **private javax.swing.JButton btnOK**
  - OK Button im Dialog.
- **private javax.swing.JButton btnAbort**
  - Beenden Button im Dialog.
- **private javax.swing.JDialog dialog**
  - Dialog.

### 5.3.6 Constructors

- **NameDialog**

**public** NameDialog(java.awt.Frame frame, data.Spieler[] spieler)

- **Description**

Hiermit wird das DialogFenster erstellt.

- **Parameters**

- \* **frame** – Das Elternfenster zu dem der Dialog gehoeren soll.
- \* **spieler** – Die Spieler.

### 5.3.7 Methods

- **cropText**

```
private void cropText(javax.swing.JTextField textField , int
    maxLenght)
```

- **Description**

Kuerzt Text in einem Textfeld auf eine maximale Laenge.

- **Parameters**

- \* **textField** – Textfeld mit den zu kuerzenden Zeichen.
- \* **maxLenght** – Maximale Zeichenlaenge.

- **showNameDialog**

```
private void showNameDialog(java.awt.Frame o)
```

- **Description**

Zeigt den Namensdialog an.

- **Parameters**

- \* **o** – Das Elternfenster zu dem der Dialog gehoeren soll.

- **verifyNames**

```
private void verifyNames()
```

- **Description**

Diese Methode ueberprueft die Eingabe in den beiden Textfeldern.

## 5.4 Class NameDialog.EnterKeyListener

Ein KeyListener fuer die Enter-Taste.

### 5.4.1 Declaration

```
class NameDialog.EnterKeyListener
extends java.awt.event.KeyAdapter
```

### 5.4.2 Field summary

**MAX\_LENGTH** Maximale Laenge der Zeichenkette bei der Namenseingabe.

### 5.4.3 Constructor summary

```
NameDialog.EnterKeyListener()
```



#### 5.4.4 Method summary

`keyPressed(KeyEvent)`  
`keyReleased(KeyEvent)`

#### 5.4.5 Fields

- `private static final int MAX_LENGTH`  
 – Maximale Laenge der Zeichenkette bei der Namenseingabe.

#### 5.4.6 Constructors

- `NameDialog.EnterKeyListener`

`NameDialog.EnterKeyListener()`

#### 5.4.7 Methods

- `keyPressed`

`void keyPressed(java.awt.event.KeyEvent arg0)`

- `keyReleased`

`void keyReleased(java.awt.event.KeyEvent arg0)`

#### 5.4.8 Members inherited from class `KeyAdapter`

`java.awt.event.KeyAdapter`

- `public void keyPressed(KeyEvent arg0)`
- `public void keyReleased(KeyEvent arg0)`
- `public void keyTyped(KeyEvent arg0)`

### 5.5 Class View

Ist fuer die Darstellung des Spiels auf dem Bildschirm verantwortlich.

#### 5.5.1 Declaration

```
public class View
    extends java.lang.Object implements java.awt.event.
        ContainerListener , ContractView
```

### 5.5.2 Field summary

**buttonListener** Der ActionListener fuer die Buttons auf dem Spielfeld.  
**buttons** Dieses Array haelt alle Buttons fuer das Spielfeld.  
**frame** Frame fuer die View.  
**lbl\_ErgebnisAusgabe** Label fuer die Ergebnisausgabe.  
**lbl\_player1Points** Label fuer die Punkte des Spieler 1.  
**lbl\_player2Points** Label fuer die Punkte des Spieler 1.  
**lbl\_PlayerName** Label fuer den Spielernamen.  
**paintingComponent** Leinwand im Hintergrund des Spielfelds.  
**repeat** True wenn das Spiel wiederholt werden soll.

### 5.5.3 Constructor summary

**View()**

### 5.5.4 Method summary

**allInactive()**  
**closeWindow()**  
**componentAdded(ContainerEvent)** Das ist eine Methode die fuer den ContainerListener implementiert werden muss.  
**componentRemoved(ContainerEvent)** Das ist eine Methode die fuer den ContainerListener implementiert werden muss.  
**fillAllButtons(Spieler)**  
**fillContainer(Container, Kasten, int)** Fuehrt einen Container mit 9 Buttons und den entsprechenden Feldnummern.  
**fuelleKasten(int, Color)**  
**generateControlElements(Spieler[], JButton, JButton, JButton, JLabel, JLabel)** Initialisiert die Steuer und Anzeigeelemente.  
**setPlayerLabel(Spieler)**  
**setRepeat(boolean)**  
**setWuerfelLabel(int)**  
**shouldRepeat()**  
**show(Kasten[], Spieler[], ActionListener, ActionListener)**  
**showRulesDialog()** Erstellt ein Dialogfenster mit den Spielregeln in Textform  
**updateButtons()**  
**updatePoints(Spieler[])**

### 5.5.5 Fields

- **private CustomButton[] buttons**
  - Dieses Array haelt alle Buttons fuer das Spielfeld.
- **private View.PaintingComponent paintingComponent**
  - Leinwand im Hintergrund des Spielfelds.
- **private javax.swing.JLabel lbl\_PlayerName**

- Label fuer den Spielernamen.
- `private javax.swing.JLabel lbl_ErgebnisAusgabe`
  - Label fuer die Ergebnisausgabe.
- `private javax.swing.JLabel lbl_player1Points`
  - Label fuer die Punkte des Spieler 1.
- `private javax.swing.JLabel lbl_player2Points`
  - Label fuer die Punkte des Spieler 1.
- `private javax.swing.JFrame frame`
  - Frame fuer die View.
- `private java.awt.event.ActionListener buttonListener`
  - Der ActionListener fuer die Buttons auf dem Spielfeld.
- `private boolean repeat`
  - True wenn das Spiel wiederholt werden soll.

#### 5.5.6 Constructors

- **View**

```
public View()
```

#### 5.5.7 Methods

- **allInactive**

```
boolean allInactive()
```

- **Description copied from ContractView** (in [5.1](#), page [25](#))  
Es wird geprueft, ob alle Buttons des Spielfeldes deaktiviert sind.
- **Returns** – Gibt true zurueck, wenn alle Buttons inaktiv sind.

- **closeWindow**

```
void closeWindow()
```

- **Description copied from ContractView** (in [5.1](#), page [25](#))  
Schliessen des Frame.

- **componentAdded**

```
public void componentAdded(java.awt.event.ContainerEvent e)
```

– **Description**

Das ist eine Methode die fuer den ContainerListener implementiert werden muss. Diese wird aufgerufen, wenn eine Komponente zum Container hinzugefuegt wird. Ist diese Komponente ein Button, wird ihm ein ActionListener zugeordnet.

– **Parameters**

\* `e` – Das zu uebergende ContainerEvent.

• **componentRemoved**

```
public void componentRemoved(java.awt.event.ContainerEvent e)
```

– **Description**

Das ist eine Methode die fuer den ContainerListener implementiert werden muss. Diese wird aufgerufen, wenn eine Komponente vom Container entfernt wird. Ist diese Komponente ein Button, wird ihm ein ActionListener entfernt.

– **Parameters**

\* `e` – Das zu uebergende ContainerEvent.

• **fillAllButtons**

```
void fillAllButtons(data.Spieler spieler)
```

– **Description copied from ContractView (in 5.1, page 25)**

Fuellen aller Felder mit der Farbe des aktuellen Spielers. Wird bei einem Gewinn benoetigt.

– **Parameters**

\* `spieler` – Aktueller Spieler.

• **fillContainer**

```
private void fillContainer(java.awt.Container c,data.Kasten
    kasten,int kastenIndex)
```

– **Description**

Fuellt einen Container mit 9 Buttons und den entsprechenden Feldnummern.

– **Parameters**

\* `c` – Der Container, der mit Buttons gefuellt werden soll.

\* `k` – Der Kasten, aus dem die Feldnummern geholt werden sollen.

\* `kastenIndex` – Der Index des Kastens

- **fuelleKasten**

```
void fuelleKasten(int kastenIndex, java.awt.Color color)
```

- **Description copied from ContractView** (in 5.1, page 25)

Alle Buttons in einem gewonnenen Kasten werden mit der Farbe des Spielers gefuellt, der ihn gewonnen hat.

- **Parameters**

- \* **kastenIndex** – - Der Index des gewonnenen Kastens.
    - \* **color** – - Die Farbe des Spieler.

- **generateControlElements**

```
private void generateControlElements(data.Spieler[] spieler ,  
    javax.swing.JButton closeButton, javax.swing.JButton  
    restartButton, javax.swing.JButton rulesButton, javax.swing.  
    JLabel lbl_wuerfelErgebnis, javax.swing.JLabel  
    lbl_activePlayer)
```

- **Description**

Initialisiert die Steuer und Anzeigeelemente.

- **Parameters**

- \* **spieler** – Array mit den Spielern.
    - \* **closeButton** – Schliessen Button.
    - \* **restartButton** – Neustart Button.
    - \* **rulesButton** – Regel Button.
    - \* **lbl\_wuerfelErgebnis** – Das Label des Wuerfelergebnis.
    - \* **lbl\_activePlayer** – Das Label des aktiven Spielers.

- **setPlayerLabel**

```
void setPlayerLabel(data.Spieler spieler)
```

- **Description copied from ContractView** (in 5.1, page 25)

Der Text des PlayerLabels wird auf den Namen des aktuellen Spieler gesetzt.

- **Parameters**

- \* **spieler** – - Der aktuelle Spieler.

- **setRepeat**

```
void setRepeat(boolean repeat)
```

- **Description copied from ContractView** (in 5.1, page 25)  
Setzt den Wunsch auf ein neues Spiel.
- **Parameters**
  - \* `repeat` – true wenn ein neues Spiel gewünscht ist. Sonst false.

- **setWuerfelLabel**

```
void setWuerfelLabel(int wuerfelErgebnis)
```

- **Description copied from ContractView** (in 5.1, page 25)  
Der Text des WuerfelLabels wird auf die aktuell gewürfelte Zahl gesetzt.
- **Parameters**
  - \* `wuerfelErgebnis` – Das aktuelle wuerfelErgebnis.

- **shouldRepeat**

```
boolean shouldRepeat()
```

- **Description copied from ContractView** (in 5.1, page 25)  
Gibt zurück ob ein neues Spiel gewünscht ist.
- **Returns** – true wenn ein neues Spiel gewünscht ist. Sonst false.

- **show**

```
void show(data.Kasten[] kaesten, data.Spieler[] spieler, java.awt.
    event.ActionListener buttonListener, java.awt.event.
    ActionListener restartListener)
```

- **Description copied from ContractView** (in 5.1, page 25)  
Initialisiert das Hauptfenster und zeigt den Dialog fuer die Namenseingabe an.
- **Parameters**
  - \* `kaesten` – Das Spielfeld.
  - \* `spieler` – Die Spieler.
  - \* `buttonListener` – Der ActionListener fuer die Buttons des Spielfeldes.
  - \* `restartListener` – Der ActionListener fuer den Neustart-Button.

- **showRulesDialog**

```
private void showRulesDialog()
```

- **Description**  
Erstellt ein Dialogfenster mit den Spielregeln in Textform

- **updateButtons**

```
void updateButtons()
```

- **Description copied from ContractView** (in 5.1, page 25)

Das Aussehen der Buttons wird aktualisiert. Nicht moegliche Felder werden deaktiviert, mgliche Felder werden hervorgehoben.

- **updatePoints**

```
void updatePoints(data.Spieler[] spieler)
```

- **Description copied from ContractView** (in 5.1, page 25)

Der aktuelle Punktestand der Spieler wird mit der View synchronisiert.

- **Parameters**

\* `spieler` – Die beiden Spieler

## 5.6 Class View.PaintingComponent

Zum Zeichnen von geometrischen Formen

### 5.6.1 Declaration

```
class View.PaintingComponent
extends javax.swing.JComponent
```

### 5.6.2 Field summary

```
color
serialVersionUID
shape
```

### 5.6.3 Constructor summary

```
View.PaintingComponent()
```

### 5.6.4 Method summary

```
paintComponent(Graphics) Darf nicht vom Programmierer aufgerufen werden!
setColor(Color) Die Farbe bestimmen
setShape(Shape) Die Form bestimmen
```

### 5.6.5 Fields

- `private static final long serialVersionUID`
- `private java.awt.Shape shape`
- `private java.awt.Color color`

### 5.6.6 Constructors

- `View.PaintingComponent`

`View.PaintingComponent()`

### 5.6.7 Methods

- `paintComponent`

`protected void paintComponent(java.awt.Graphics g)`

- **Description**

Darf nicht vom Programmierer aufgerufen werden! Java ruft diese Methode bei bedarf selbst auf.

- `setColor`

`public void setColor(java.awt.Color color)`

- **Description**

Die Farbe bestimmen

- **Parameters**

\* `color` – - Die Farbe, die das Objekt haben soll

- `setShape`

`public void setShape(java.awt.Shape shape)`

- **Description**

Die Form bestimmen

- **Parameters**

\* `shape` – - Die geometrische Grundform



## 5.6.8 Members inherited from class JComponent

javax.swing.JComponent

- void **\_paintImmediately**(int arg0, int arg1, int arg2, int arg3)
- static void **()**
- private transient **aaTextInfo**
- protected **accessibleContext**
- private **actionMap**
- private static final **ACTIONMAP\_CREATED**
- public void **addAncestorListener**(event.AncestorListener arg0)
- public void **addNotify**()
- public synchronized void **addVetoableChangeListener**(java.beans.VetoableChangeListener arg0)
- private void **adjustPaintFlags**()
- private **alignmentX**
- private **alignmentY**
- boolean **alwaysOnTop**()
- private static final **ANCESTOR\_INPUTMAP\_CREATED**
- private static final **ANCESTOR\_USING\_BUFFER**
- private **ancestorInputMap**
- private **autoscrolls**
- private static final **AUTOSCROLLS\_SET**
- private **border**
- boolean **checkIfChildObscuredBySibling**()
- private transient **clientProperties**
- void **clientPropertyChanged**(java.lang.Object arg0, java.lang.Object arg1, java.lang.Object arg2)
- private static final **COMPLETELY\_OBSCURED**
- void **componentInputMapChanged**(ComponentInputMap arg0)
- private static **componentObtainingGraphicsFrom**
- private static **componentObtainingGraphicsFromLock**
- static final void **computeVisibleRect**(java.awt.Component arg0, java.awt.Rectangle arg1)
- public void **computeVisibleRect**(java.awt.Rectangle arg0)
- void **compWriteObjectNotify**()
- public boolean **contains**(int arg0, int arg1)
- private static final **CREATED\_DOUBLE\_BUFFER**
- public JToolTip **createToolTip**()
- static **DEBUG\_GRAPHICS\_LOADED**
- private static final **defaultLocale**
- private void **deregisterNextFocusableComponent**()
- public void **disable**()
- void **dndDone**()
- TransferHandler.DropLocation **dropLocationForPoint**(java.awt.Point arg0)
- public void **enable**()
- private static Rectangle **fetchRectangle**()
- public void **firePropertyChange**(java.lang.String arg0, boolean arg1, boolean arg2)
- public void **firePropertyChange**(java.lang.String arg0, char arg1, char arg2)
- public void **firePropertyChange**(java.lang.String arg0, int arg1, int arg2)
- protected void **fireVetoableChange**(java.lang.String arg0, java.lang.Object arg1, java.lang.Object arg2) throws java.beans.PropertyVetoException
- private **flags**
- private static final **FOCUS\_INPUTMAP\_CREATED**
- private static final **FOCUS\_TRAVERSAL\_KEYS\_BACKWARD\_SET**
- private static final **FOCUS\_TRAVERSAL\_KEYS\_FORWARD\_SET**

- `static final focusController`
- `private focusInputMap`
- `public AccessibleContext getAccessibleContext()`
- `public ActionListener getActionForKeyStroke(KeyStroke arg0)`
- `public final ActionMap getActionMap()`
- `final ActionMap getActionMap(boolean arg0)`
- `public float getAlignmentX()`
- `public float getAlignmentY()`
- `public AncestorListener getAncestorListeners()`
- `private AncestorNotifier getAncestorNotifier()`
- `public boolean getAutoscrolls()`
- `public int getBaseline(int arg0, int arg1)`
- `public Component.BaselineResizeBehavior getBaselineResizeBehavior()`
- `public Border getBorder()`
- `public Rectangle getBounds(java.awt.Rectangle arg0)`
- `private ArrayTable getClientProperties()`
- `public final Object getClientProperty(java.lang.Object arg0)`
- `protected Graphics getComponentGraphics(java.awt.Graphics arg0)`
- `public JPopupMenu getComponentPopupMenu()`
- `public int getConditionForKeyStroke(KeyStroke arg0)`
- `boolean getCreatedDoubleBuffer()`
- `public int getDebugGraphicsOptions()`
- `public static Locale getDefaultLocale()`
- `private boolean getFlag(int arg0)`
- `public FontMetrics getFontMetrics(java.awt.Font arg0)`
- `public Graphics getGraphics()`
- `static void getGraphicsInvoked(java.awt.Component arg0)`
- `public int getHeight()`
- `public boolean getInheritsPopupMenu()`
- `public final InputMap getInputMap()`
- `public final InputMap getInputMap(int arg0)`
- `final InputMap getInputMap(int arg0, boolean arg1)`
- `public InputVerifier getInputVerifier()`
- `public Insets getInsets()`
- `public Insets getInsets(java.awt.Insets arg0)`
- `public EventListener getListeners(java.lang.Class arg0)`
- `public Point getLocation(java.awt.Point arg0)`
- `static Set getManagingFocusBackwardTraversalKeys()`
- `static Set getManagingFocusForwardTraversalKeys()`
- `public Dimension getMaximumSize()`
- `public Dimension getMinimumSize()`
- `public Component getNextFocusableComponent()`
- `private int getObscuredState(int arg0, int arg1, int arg2, int arg3, int arg4)`
- `public Point getPopupLocation(java.awt.event.MouseEvent arg0)`
- `public Dimension getPreferredSize()`
- `public KeyStroke getRegisteredKeyStrokes()`
- `public JRootPane getRootPane()`
- `public Dimension getSize(java.awt.Dimension arg0)`
- `public Point getToolTipLocation(java.awt.event.MouseEvent arg0)`
- `public String getToolTipText()`
- `public String getToolTipText(java.awt.event.MouseEvent arg0)`
- `public Container getTopLevelAncestor()`
- `public TransferHandler getTransferHandler()`
- `public String getUIClassID()`
- `public boolean getVerifyInputWhenFocusTarget()`
- `public synchronized VetoableChangeListener getVetoableChangeListeners()`

- public Rectangle getVisibleRect()
- public int getWidth()
- static byte getWriteObjCounter(JComponent arg0)
- public int getX()
- public int getY()
- public void grabFocus()
- private static final INHERITS\_POPUP\_MENU
- private static final INPUT\_VERIFIER\_SOURCE\_KEY
- private inputVerifier
- private static final IS\_DOUBLE\_BUFFERED
- private static final IS\_OPAQUE
- private static final IS\_PAINTING\_TILE
- private static final IS\_PRINTING
- private static final IS\_PRINTING\_ALL
- private static final IS\_REPAINTING
- private isAlignmentXSet
- private isAlignmentYSet
- private static boolean isComponentObtainingGraphicsFrom(java.awt.Component arg0)
- public boolean isDoubleBuffered()
- public static boolean isLightweightComponent(java.awt.Component arg0)
- public boolean isManagingFocus()
- public boolean isOpaque()
- public boolean isOptimizedDrawingEnabled()
- boolean isPainting()
- public final boolean isPaintingForPrint()
- protected boolean isPaintingOrigin()
- public boolean isPaintingTile()
- public boolean isRequestFocusEnabled()
- public boolean isValidRoot()
- private static final KEY\_EVENTS\_ENABLED
- private static final KEYBOARD\_BINDINGS\_KEY
- protected listenerList
- private static managingFocusBackwardTraversalKeys
- private static managingFocusForwardTraversalKeys
- private static final NEXT\_FOCUS
- private static final NOT\_OBSCURED
- private static final OPAQUE\_SET
- public void paint(java.awt.Graphics arg0)
- protected void paintBorder(java.awt.Graphics arg0)
- protected void paintChildren(java.awt.Graphics arg0)
- protected void paintComponent(java.awt.Graphics arg0)
- void paintForceDoubleBuffered(java.awt.Graphics arg0)
- public void paintImmediately(int arg0, int arg1, int arg2, int arg3)
- public void paintImmediately(java.awt.Rectangle arg0)
- transient paintingChild
- void paintToOffscreen(java.awt.Graphics arg0, int arg1, int arg2, int arg3, int arg4, int arg5, int arg6)
- protected String paramString()
- private static final PARTIALLY\_OBSCURED
- private popupMenu
- public void print(java.awt.Graphics arg0)
- public void printAll(java.awt.Graphics arg0)
- protected void printBorder(java.awt.Graphics arg0)
- protected void printChildren(java.awt.Graphics arg0)
- protected void printComponent(java.awt.Graphics arg0)

- protected void **processComponentKeyEvent**(java.awt.event.KeyEvent arg0)
- protected boolean **processKeyBinding**(KeyStroke arg0, java.awt.event.KeyEvent arg1, int arg2, boolean arg3)
- boolean **processKeyBindings**(java.awt.event.KeyEvent arg0, boolean arg1)
- static boolean **processKeyBindingsForAllComponents**(java.awt.event.KeyEvent arg0, java.awt.Container arg1, boolean arg2)
- protected void **processKeyEvent**(java.awt.event.KeyEvent arg0)
- protected void **processMouseEvent**(java.awt.event.MouseEvent arg0)
- protected void **processMouseMotionEvent**(java.awt.event.MouseEvent arg0)
- public final void **putClientProperty**(java.lang.Object arg0, java.lang.Object arg1)
- private void **readObject**(java.io.ObjectInputStream arg0) throws java.io.IOException, java.lang.ClassNotFoundException
- private static final **readObjectCallbacks**
- boolean **rectangleIsObscured**(int arg0, int arg1, int arg2, int arg3)
- private static void **recycleRectangle**(java.awt.Rectangle arg0)
- public void **registerKeyboardAction**(java.awt.event.ActionListener arg0, KeyStroke arg1, int arg2)
- public void **registerKeyboardAction**(java.awt.event.ActionListener arg0, java.lang.String arg1, KeyStroke arg2, int arg3)
- private void **registerNextFocusableComponent**()
- private void **registerNextFocusableComponent**(java.awt.Component arg0)
- private void **registerWithKeyboardManager**(boolean arg0)
- private void **registerWithKeyboardManager**(KeyStroke arg0)
- public void **removeAncestorListener**(event.AncestorListener arg0)
- public void **removeNotify**()
- public synchronized void **removeVetoableChangeListener**(java.beans.VetoableChangeListener arg0)
- public void **repaint**(long arg0, int arg1, int arg2, int arg3, int arg4)
- public void **repaint**(java.awt.Rectangle arg0)
- private static final **REQUEST\_FOCUS\_DISABLED**
- public boolean **requestDefaultFocus**()
- public void **requestFocus**()
- public boolean **requestFocus**(boolean arg0)
- public boolean **requestFocusInWindow**()
- protected boolean **requestFocusInWindow**(boolean arg0)
- private static final **RESERVED\_1**
- private static final **RESERVED\_2**
- private static final **RESERVED\_3**
- private static final **RESERVED\_4**
- private static final **RESERVED\_5**
- private static final **RESERVED\_6**
- public void **resetKeyboardActions**()
- public void **reshape**(int arg0, int arg1, int arg2, int arg3)
- private static final **REVALIDATE\_RUNNABLE\_SCHEDULED**
- public void **revalidate**()
- static Graphics **safelyGetGraphics**(java.awt.Component arg0)
- static Graphics **safelyGetGraphics**(java.awt.Component arg0, java.awt.Component arg1)
- public void **scrollRectToVisible**(java.awt.Rectangle arg0)
- public final void **setActionMap**(ActionMap arg0)
- public void **setAlignmentX**(float arg0)
- public void **setAlignmentY**(float arg0)
- public void **setAutoscrolls**(boolean arg0)
- public void **setBackground**(java.awt.Color arg0)
- public void **setBorder**(border.Border arg0)

- `public void setComponentPopupMenu(JPopupMenu arg0)`
- `void setCreatedDoubleBuffer(boolean arg0)`
- `public void setDebugGraphicsOptions(int arg0)`
- `public static void setDefaultLocale(java.util.Locale arg0)`
- `public void setDoubleBuffered(boolean arg0)`
- `Object setDropLocation(TransferHandler.DropLocation arg0, java.lang.Object arg1, boolean arg2)`
- `public void setEnabled(boolean arg0)`
- `private void setFlag(int arg0, boolean arg1)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFont(java.awt.Font arg0)`
- `public void setForeground(java.awt.Color arg0)`
- `public void setInheritsPopupMenu(boolean arg0)`
- `public final void setInputMap(int arg0, InputMap arg1)`
- `public void setInputVerifier(InputVerifier arg0)`
- `public void setMaximumSize(java.awt.Dimension arg0)`
- `public void setMinimumSize(java.awt.Dimension arg0)`
- `public void setNextFocusableComponent(java.awt.Component arg0)`
- `public void setOpaque(boolean arg0)`
- `void setPaintingChild(java.awt.Component arg0)`
- `public void setPreferredSize(java.awt.Dimension arg0)`
- `public void setRequestFocusEnabled(boolean arg0)`
- `public void setToolTipText(java.lang.String arg0)`
- `public void setTransferHandler(TransferHandler arg0)`
- `protected void setUI(plaf.ComponentUI arg0)`
- `void setUIProperty(java.lang.String arg0, java.lang.Object arg1)`
- `public void setVerifyInputWhenFocusTarget(boolean arg0)`
- `public void setVisible(boolean arg0)`
- `static void setWriteObjCounter(JComponent arg0, byte arg1)`
- `int shouldDebugGraphics()`
- `void superProcessMouseEvent(java.awt.event.MouseEvent arg0)`
- `private static tempRectangles`
- `public static final TOOL_TIP_TEXT_KEY`
- `protected transient ui`
- `private static final uiClassID`
- `public static final UNDEFINED_CONDITION`
- `private void uninstallUIAndProperties()`
- `public void unregisterKeyboardAction(KeyStroke arg0)`
- `private void unregisterWithKeyboardManager()`
- `private void unregisterWithKeyboardManager(KeyStroke arg0)`
- `public void update(java.awt.Graphics arg0)`
- `public void updateUI()`
- `private verifyInputWhenFocusTarget`
- `private vetoableChangeSupport`
- `public static final WHEN_ANCESTOR_OF_FOCUSED_COMPONENT`
- `public static final WHEN_FOCUSED`
- `public static final WHEN_IN_FOCUSED_WINDOW`
- `private static final WHEN_IN_FOCUSED_WINDOW_BINDINGS`
- `private static final WIF_INPUTMAP_CREATED`
- `private windowInputMap`
- `private static final WRITE_OBJ_COUNTER_FIRST`
- `private static final WRITE_OBJ_COUNTER_LAST`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`

## 5.6.9 Members inherited from class Container

```
java.awt.Container
```

- static void ()
- public Component add(Component arg0)
- public Component add(Component arg0, int arg1)
- public void add(Component arg0, java.lang.Object arg1)
- public void add(Component arg0, java.lang.Object arg1, int arg2)
- public Component add(java.lang.String arg0, Component arg1)
- public synchronized void addContainerListener(event.ContainerListener arg0)
- private void addDelicately(Component arg0, Container arg1, int arg2)
- protected void addImpl(Component arg0, java.lang.Object arg1, int arg2)
- public void addNotify()
- public void addPropertyChangeListener(java.beans.PropertyChangeListener arg0)
- public void addPropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)
- void adjustDecendantsOnParent(int arg0)
- void adjustDescendants(int arg0)
- void adjustListeningChildren(long arg0, int arg1)
- public void applyComponentOrientation(ComponentOrientation arg0)
- public boolean areFocusTraversalKeysSet(int arg0)
- boolean canContainFocusOwner(Component arg0)
- private void checkAdding(Component arg0, int arg1)
- private void checkAddToSelf(Component arg0)
- void checkGD(java.lang.String arg0)
- private void checkNotAWindow(Component arg0)
- void clearCurrentFocusCycleRootOnHide()
- void clearMostRecentFocusOwnerOnHide()
- private component
- transient containerListener
- private containerSerializedDataVersion
- final boolean containsFocus()
- public int countComponents()
- int countHierarchyMembers()
- final void createChildHierarchyEvents(int arg0, long arg1, boolean arg2)
- final int createHierarchyEvents(int arg0, Component arg1, Container arg2, long arg3, boolean arg4)
- final void decreaseComponentCount(Component arg0)
- public void deliverEvent(Event arg0)
- transient descendantsCount
- private static descendUnconditionallyWhenValidating
- private dispatcher
- void dispatchEventImpl(AWTEvent arg0)
- void dispatchEventToSelf(AWTEvent arg0)
- public void doLayout()
- private static final EMPTY\_ARRAY
- boolean eventEnabled(AWTEvent arg0)
- private static final eventLog
- public Component findComponentAt(int arg0, int arg1)
- final Component findComponentAt(int arg0, int arg1, boolean arg2)
- public Component findComponentAt(Point arg0)
- final Component findComponentAtImpl(int arg0, int arg1, boolean arg2)
- private Container findTraversalRoot()
- private focusCycleRoot
- private transient focusTraversalPolicy
- private focusTraversalPolicyProvider

- Accessible getAccessibleAt(Point arg0)
- Accessible getAccessibleChild(int arg0)
- int getAccessibleChildrenCount()
- public float getAlignmentX()
- public float getAlignmentY()
- private int getBottommostComponentIndex()
- public Component getComponent(int arg0)
- public Component getComponentAt(int arg0, int arg1)
- public Component getComponentAt(Point arg0)
- public int getComponentCount()
- final Component getComponents\_NoClientCode()
- public Component getComponents()
- Component getComponentsSync()
- public int getComponentZOrder(Component arg0)
- public synchronized ContainerListener getContainerListeners()
- Component getDropTargetEventTarget(int arg0, int arg1, boolean arg2)
- public Set getFocusTraversalKeys(int arg0)
- public FocusTraversalPolicy getFocusTraversalPolicy()
- Container getHeavyweightContainer()
- public Insets getInsets()
- public LayoutManager getLayout()
- public EventListener getListeners(java.lang.Class arg0)
- private int getListenersCount(int arg0, boolean arg1)
- public Dimension getMaximumSize()
- public Dimension getMinimumSize()
- Component getMouseEventTarget(int arg0, int arg1, boolean arg2)
- private Component getMouseEventTarget(int arg0, int arg1, boolean arg2, Container.EventTargetFilter arg3, boolean arg4)
- private Component getMouseEventTargetImpl(int arg0, int arg1, boolean arg2, Container.EventTargetFilter arg3, boolean arg4, boolean arg5)
- public Point getMousePosition(boolean arg0) throws HeadlessException
- final Region getOpaqueShape()
- public Dimension getPreferredSize()
- private int getTopmostComponentIndex()
- final Container getTraversalRoot()
- final boolean hasHeavyweightDescendants()
- final boolean hasLightweightDescendants()
- static final INCLUDE\_SELF
- final void increaseComponentCount(Component arg0)
- void initializeFocusTraversalKeys()
- private static native void initIDs()
- public Insets insets()
- public void invalidate()
- void invalidateParent()
- void invalidateTree()
- public boolean isAncestorOf(Component arg0)
- public boolean isFocusCycleRoot()
- public boolean isFocusCycleRoot(Container arg0)
- public final boolean isFocusTraversalPolicyProvider()
- public boolean isFocusTraversalPolicySet()
- private static final isJavaAwtSmartInvalidate
- private boolean isParentOf(Component arg0)
- final boolean isRecursivelyVisibleUpToHeavyweightContainer()
- private static boolean isRemoveNotifyNeeded(Component arg0, Container arg1, Container arg2)
- boolean isSameOrAncestorOf(Component arg0, boolean arg1)

- `public boolean isValidRoot()`
- `public void layout()`
- `layoutMgr`
- `void lightweightPaint(Graphics arg0)`
- `void lightweightPrint(Graphics arg0)`
- `public void list(java.io.PrintStream arg0, int arg1)`
- `public void list(java.io.PrintWriter arg0, int arg1)`
- `transient listeningBoundsChildren`
- `transient listeningChildren`
- `public Component locate(int arg0, int arg1)`
- `private static final log`
- `public Dimension minimumSize()`
- `private static final mixingLog`
- `void mixOnHiding(boolean arg0)`
- `void mixOnReshaping()`
- `void mixOnShowing()`
- `void mixOnValidating()`
- `void mixOnZOrderChanging(int arg0, int arg1)`
- `transient modalAppContext`
- `transient modalComp`
- `int numListening(long arg0)`
- `private transient numOfHWComponents`
- `private transient numOfLWComponents`
- `public void paint(Graphics arg0)`
- `public void paintComponents(Graphics arg0)`
- `void paintHeavyweightComponents(Graphics arg0)`
- `protected String paramString()`
- `void postProcessKeyEvent(event.KeyEvent arg0)`
- `boolean postsOldMouseEvents()`
- `public Dimension preferredSize()`
- `void preProcessKeyEvent(event.KeyEvent arg0)`
- `transient preserveBackgroundColor`
- `public void print(Graphics arg0)`
- `public void printComponents(Graphics arg0)`
- `void printHeavyweightComponents(Graphics arg0)`
- `private transient printing`
- `private transient printingThreads`
- `protected void processContainerEvent(event.ContainerEvent arg0)`
- `protected void processEvent(AWTEvent arg0)`
- `void proxyEnableEvents(long arg0)`
- `private void readObject(java.io.ObjectInputStream arg0) throws java.lang.ClassNotFoundException, java.io.IOException`
- `final void recursiveApplyCurrentShape()`
- `final void recursiveApplyCurrentShape(int arg0)`
- `final void recursiveApplyCurrentShape(int arg0, int arg1)`
- `private void recursiveHideHeavyweightChildren()`
- `private void recursiveRelocateHeavyweightChildren(Point arg0)`
- `private void recursiveShowHeavyweightChildren()`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0)`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0, int arg1)`
- `final void recursiveSubtractAndApplyShape(sun.java2d.pipe.Region arg0, int arg1, int arg2)`
- `public void remove(Component arg0)`
- `public void remove(int arg0)`
- `public void removeAll()`



- `public synchronized void removeContainerListener(event.ContainerListener arg0)`
- `private boolean removeDelicately(Component arg0, Container arg1, int arg2)`
- `public void removeNotify()`
- `private void reparentChild(Component arg0)`
- `private void reparentTraverse(peer.ContainerPeer arg0, Container arg1)`
- `static final SEARCH_HEAVYWEIGHTS`
- `private static final serialPersistentFields`
- `private static final serialVersionUID`
- `public void setComponentZOrder(Component arg0, int arg1)`
- `public void setFocusCycleRoot(boolean arg0)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalPolicy(FocusTraversalPolicy arg0)`
- `public final void setFocusTraversalPolicyProvider(boolean arg0)`
- `public void setFont(Font arg0)`
- `public void setLayout(LayoutManager arg0)`
- `private void startLWModal()`
- `private void stopLWModal()`
- `public void transferFocusDownCycle()`
- `public void update(Graphics arg0)`
- `boolean updateGraphicsData(GraphicsConfiguration arg0)`
- `public void validate()`
- `protected void validateTree()`
- `final void validateUnconditionally()`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`

### 5.6.10 Members inherited from class Component

`java.awt.Component`

- `static void ()`
- `private transient volatile acc`
- `accessibleContext`
- `public boolean action(Event arg0, java.lang.Object arg1)`
- `static final ActionListenerK`
- `public void add(PopupMenu arg0)`
- `public synchronized void addComponentListener(event.ComponentListener arg0)`
- `public synchronized void addFocusListener(event.FocusListener arg0)`
- `public void addHierarchyBoundsListener(event.HierarchyBoundsListener arg0)`
- `public void addHierarchyListener(event.HierarchyListener arg0)`
- `public synchronized void addInputMethodListener(event.InputMethodListener arg0)`
- `public synchronized void addKeyListener(event.KeyListener arg0)`
- `public synchronized void addMouseListener(event.MouseListener arg0)`
- `public synchronized void addMouseMotionListener(event.MouseMotionListener arg0)`
- `public synchronized void addMouseWheelListener(event.MouseWheelListener arg0)`
- `public void addNotify()`
- `public void addPropertyChangeListener(java.beans.PropertyChangeListener arg0)`
- `public void addPropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)`
- `void adjustListeningChildrenOnParent(long arg0, int arg1)`
- `static final adjustmentListenerK`
- `transient appContext`
- `public void applyComponentOrientation(ComponentOrientation arg0)`
- `void applyCompoundShape(sun.java2d.pipe.Region arg0)`

- `void applyCurrentShape()`
- `private final void applyCurrentShapeBelowMe()`
- `final boolean areBoundsValid()`
- `public boolean areFocusTraversalKeysSet(int arg0)`
- `boolean areInputMethodsEnabled()`
- `private autoFocusTransferOnDisposal`
- `void autoProcessMouseWheel(event.MouseWheelEvent arg0)`
- `background`
- `transient backgroundEraseDisabled`
- `public static final BOTTOM_ALIGNMENT`
- `public Rectangle bounds()`
- `private boundsOp`
- `transient bufferStrategy`
- `private Region calculateCurrentShape()`
- `final boolean canBeFocusOwner()`
- `final boolean canBeFocusOwnerRecursively()`
- `public static final CENTER_ALIGNMENT`
- `private changeSupport`
- `private boolean checkCoalescing()`
- `void checkGD(java.lang.String arg0)`
- `public int checkImage(Image arg0, image.ImageObserver arg1)`
- `public int checkImage(Image arg0, int arg1, int arg2, image.ImageObserver arg3)`
- `final void checkTreeLock()`
- `boolean checkWindowClosingException()`
- `void clearCurrentFocusCycleRootOnHide()`
- `void clearMostRecentFocusOwnerOnHide()`
- `protected AWTEvent coalesceEvents(AWTEvent arg0, AWTEvent arg1)`
- `private static final coalesceEventsParams`
- `private static final coalesceMap`
- `private transient coalescingEnabled`
- `transient componentListener`
- `static final componentListenerK`
- `transient componentOrientation`
- `private componentSerializedDataVersion`
- `private transient compoundShape`
- `String constructComponentName()`
- `static final containerListenerK`
- `public boolean contains(int arg0, int arg1)`
- `public boolean contains(Point arg0)`
- `boolean containsFocus()`
- `int countHierarchyMembers()`
- `void createBufferStrategy(int arg0)`
- `void createBufferStrategy(int arg0, BufferCapabilities arg1) throws AWTException`
- `int createHierarchyEvents(int arg0, Component arg1, Container arg2, long arg3, boolean arg4)`
- `public Image createImage(image.ImageProducer arg0)`
- `public Image createImage(int arg0, int arg1)`
- `public VolatileImage createVolatileImage(int arg0, int arg1)`
- `public VolatileImage createVolatileImage(int arg0, int arg1, ImageCapabilities arg2) throws AWTException`
- `cursor`
- `public void deliverEvent(Event arg0)`
- `public void disable()`
- `protected final void disableEvents(long arg0)`
- `public final void dispatchEvent(AWTEvent arg0)`

- `void dispatchEventImpl(AWTEvent arg0)`
- `boolean dispatchMouseWheelToAncestor(event.MouseWheelEvent arg0)`
- `public void doLayout()`
- `private void doSwingSerialization()`
- `dropTarget`
- `public void enable()`
- `public void enable(boolean arg0)`
- `enabled`
- `protected final void enableEvents(long arg0)`
- `public void enableInputMethods(boolean arg0)`
- `transient eventCache`
- `boolean eventEnabled(AWTEvent arg0)`
- `private static final eventLog`
- `eventMask`
- `boolean eventTypeEnabled(int arg0)`
- `Component findUnderMouseInWindow(PointerInfo arg0)`
- `protected void firePropertyChange(java.lang.String arg0, boolean arg1, boolean arg2)`
- `public void firePropertyChange(java.lang.String arg0, byte arg1, byte arg2)`
- `public void firePropertyChange(java.lang.String arg0, char arg1, char arg2)`
- `public void firePropertyChange(java.lang.String arg0, double arg1, double arg2)`
- `public void firePropertyChange(java.lang.String arg0, float arg1, float arg2)`
- `protected void firePropertyChange(java.lang.String arg0, int arg1, int arg2)`
- `public void firePropertyChange(java.lang.String arg0, long arg1, long arg2)`
- `protected void firePropertyChange(java.lang.String arg0, java.lang.Object arg1, java.lang.Object arg2)`
- `public void firePropertyChange(java.lang.String arg0, short arg1, short arg2)`
- `private static final FOCUS_TRAVERSABLE_DEFAULT`
- `private static final FOCUS_TRAVERSABLE_SET`
- `private static final FOCUS_TRAVERSABLE_UNKNOWN`
- `private focusable`
- `transient focusListener`
- `static final focusListenerK`
- `private static final focusLog`
- `private static final focusTraversalKeyPropertyNames`
- `focusTraversalKeys`
- `private focusTraversalKeysEnabled`
- `font`
- `foreground`
- `final AccessControlContext getAccessControlContext()`
- `public AccessibleContext getAccessibleContext()`
- `int getAccessibleIndexInParent()`
- `AccessibleStateSet getAccessibleStateSet()`
- `public float getAlignmentX()`
- `public float getAlignmentY()`
- `private Region getAppliedShape()`
- `Image getBackBuffer()`
- `public Color getBackground()`
- `public int getBaseline(int arg0, int arg1)`
- `public Component.BaselineResizeBehavior getBaselineResizeBehavior()`
- `public Rectangle getBounds()`
- `public Rectangle getBounds(Rectangle arg0)`
- `int getBoundsOp()`
- `BufferStrategy getBufferStrategy()`
- `public ColorModel getColorModel()`
- `public Component getComponentAt(int arg0, int arg1)`

- `public Component getComponentAt(Point arg0)`
- `public synchronized ComponentListener getComponentListeners()`
- `public ComponentOrientation getComponentOrientation()`
- `Container getContainer()`
- `Window getContainingWindow()`
- `final Cursor getCursor_NoClientCode()`
- `public Cursor getCursor()`
- `public synchronized DropTarget getDropTarget()`
- `public Container getFocusCycleRootAncestor()`
- `public synchronized FocusListener getFocusListeners()`
- `final Set getFocusTraversalKeys_NoIDCheck(int arg0)`
- `public Set getFocusTraversalKeys(int arg0)`
- `public boolean getFocusTraversalKeysEnabled()`
- `final Font getFont_NoClientCode()`
- `public Font getFont()`
- `public FontMetrics getFontMetrics(Font arg0)`
- `public Color getForeground()`
- `final Graphics getGraphics_NoClientCode()`
- `public Graphics getGraphics()`
- `final GraphicsConfiguration getGraphicsConfiguration_NoClientCode()`
- `public GraphicsConfiguration getGraphicsConfiguration()`
- `public int getHeight()`
- `public synchronized HierarchyBoundsListener getHierarchyBoundsListeners()`
- `public synchronized HierarchyListener getHierarchyListeners()`
- `final ComponentPeer getHWPeeAboveMe()`
- `public boolean getIgnoreRepaint()`
- `public InputContext getInputContext()`
- `public synchronized InputMethodListener getInputMethodListeners()`
- `public InputMethodRequests getInputMethodRequests()`
- `private Insets getInsets_NoClientCode()`
- `public synchronized KeyListener getKeyListeners()`
- `public EventListener getListeners(java.lang.Class arg0)`
- `public Locale getLocale()`
- `public Point getLocation()`
- `public Point getLocation(Point arg0)`
- `final Point getLocationOnScreen_NoTreeLock()`
- `public Point getLocationOnScreen()`
- `Point getLocationOnWindow()`
- `public Dimension getMaximumSize()`
- `public Dimension getMinimumSize()`
- `public synchronized MouseListener getMouseListeners()`
- `public synchronized MouseMotionListener getMouseMotionListeners()`
- `public Point getMousePosition() throws HeadlessException`
- `public synchronized MouseWheelListener getMouseWheelListeners()`
- `public String getName()`
- `Container getNativeContainer()`
- `final Component getNextFocusCandidate()`
- `final Region getNormalShape()`
- `Object getObjectLock()`
- `Region getOpaqueShape()`
- `final Container getParent_NoClientCode()`
- `public Container getParent()`
- `public ComponentPeer getPeer()`
- `public Dimension getPreferredSize()`
- `public PropertyChangeListener getPropertyChangeListeners()`
- `public PropertyChangeListener getPropertyChangeListeners(java.lang.String arg0)`

- final int `getSiblingIndexAbove()`
- final int `getSiblingIndexBelow()`
- public Dimension `getSize()`
- public Dimension `getSize(Dimension arg0)`
- public Toolkit `getToolkit()`
- final Toolkit `getToolkitImpl()`
- Container `getTraversalRoot()`
- public final Object `getTreeLock()`
- public int `getWidth()`
- public int `getX()`
- public int `getY()`
- public boolean `gotFocus(Event arg0, java.lang.Object arg1)`
- private transient `graphicsConfig`
- public boolean `handleEvent(Event arg0)`
- public boolean `hasFocus()`
- height
- public void `hide()`
- transient `hierarchyBoundsListener`
- static final `hierarchyBoundsListenerK`
- transient `hierarchyListener`
- static final `hierarchyListenerK`
- ignoreRepaint
- public boolean `imageUpdate(Image arg0, int arg1, int arg2, int arg3, int arg4, int arg5)`
- static `incRate`
- void `initializeFocusTraversalKeys()`
- private static native void `initIDs()`
- transient `inputMethodListener`
- static final `inputMethodListenerK`
- public boolean `inside(int arg0, int arg1)`
- public void `invalidate()`
- final void `invalidateIfValid()`
- void `invalidateParent()`
- private transient `isAddNotifyComplete`
- boolean `isAutoFocusTransferOnDisposal()`
- public boolean `isBackgroundSet()`
- private static boolean `isCoalesceEventsOverridden(java.lang.Class arg0)`
- final boolean `isCoalescingEnabled()`
- public boolean `isCursorSet()`
- public boolean `isDisplayable()`
- public boolean `isDoubleBuffered()`
- public boolean `isEnabled()`
- final boolean `isEnabledImpl()`
- public boolean `isFocusable()`
- public boolean `isFocusCycleRoot(Container arg0)`
- public boolean `isFocusOwner()`
- public boolean `isFocusTraversable()`
- private `isFocusTraversableOverridden`
- final boolean `isFocusTraversableOverridden()`
- public boolean `isFontSet()`
- public boolean `isForegroundSet()`
- static `isInc`
- static boolean `isInstanceOf(java.lang.Object arg0, java.lang.String arg1)`
- public boolean `isLightweight()`
- public boolean `isMaximumSizeSet()`
- public boolean `isMinimumSizeSet()`

- final boolean isMixingNeeded()
- final boolean isNonOpaqueForMixing()
- public boolean isOpaque()
- isPacked
- public boolean isPreferredSizeSet()
- boolean isRecursivelyVisible()
- private boolean isRequestFocusAccepted(boolean arg0, boolean arg1, sun.awt.CausedFocusEvent.Cause arg2)
- boolean isSameOrAncestorOf(Component arg0, boolean arg1)
- public boolean isShowing()
- public boolean isValid()
- final boolean isVisible\_NoClientCode()
- public boolean isVisible()
- static final itemListenerK
- public boolean keyDown(Event arg0, int arg1)
- transient keyListener
- static final keyListenerK
- public boolean keyUp(Event arg0, int arg1)
- public void layout()
- public static final LEFT\_ALIGNMENT
- void lightweightPaint(Graphics arg0)
- void lightweightPrint(Graphics arg0)
- public void list()
- public void list(java.io.PrintStream arg0)
- public void list(java.io.PrintStream arg0, int arg1)
- public void list(java.io.PrintWriter arg0)
- public void list(java.io.PrintWriter arg0, int arg1)
- locale
- public Component locate(int arg0, int arg1)
- private Point location\_NoClientCode()
- public Point location()
- static final LOCK
- private static final log
- public boolean lostFocus(Event arg0, java.lang.Object arg1)
- maxSize
- maxSizeSet
- public Dimension minimumSize()
- minSize
- minSizeSet
- private transient mixingCutoutRegion
- private static final mixingLog
- void mixOnHiding(boolean arg0)
- void mixOnReshaping()
- void mixOnShowing()
- void mixOnValidating()
- void mixOnZOrderChanging(int arg0, int arg1)
- public boolean mouseDown(Event arg0, int arg1, int arg2)
- public boolean mouseDrag(Event arg0, int arg1, int arg2)
- public boolean mouseEnter(Event arg0, int arg1, int arg2)
- public boolean mouseExit(Event arg0, int arg1, int arg2)
- transient mouseListener
- static final mouseListenerK
- transient mouseMotionListener
- static final mouseMotionListenerK
- public boolean mouseMove(Event arg0, int arg1, int arg2)
- public boolean mouseUp(Event arg0, int arg1, int arg2)

- transient `mouseWheelListener`
- static final `mouseWheelListenerK`
- public void `move(int arg0, int arg1)`
- private `name`
- private `nameExplicitlySet`
- `newEventsOnly`
- public void `nextFocus()`
- private void `notifyNewBounds(boolean arg0, boolean arg1)`
- int `numListening(long arg0)`
- private transient `objectLock`
- static final `ownedWindowK`
- public void `paint(Graphics arg0)`
- public void `paintAll(Graphics arg0)`
- void `paintHeavyweightComponents(Graphics arg0)`
- protected String `paramString()`
- transient `parent`
- transient `peer`
- `peerFont`
- Point `pointRelativeToComponent(Point arg0)`
- `popups`
- public boolean `postEvent(Event arg0)`
- boolean `postsOldMouseEvents()`
- public Dimension `preferredSize()`
- `prefSize`
- `prefSizeSet`
- public boolean `prepareImage(Image arg0, image.ImageObserver arg1)`
- public boolean `prepareImage(Image arg0, int arg1, int arg2, image.ImageObserver arg3)`
- public void `print(Graphics arg0)`
- public void `printAll(Graphics arg0)`
- void `printHeavyweightComponents(Graphics arg0)`
- protected void `processComponentEvent(event.ComponentEvent arg0)`
- protected void `processEvent(AWTEvent arg0)`
- protected void `processFocusEvent(event.FocusEvent arg0)`
- protected void `processHierarchyBoundsEvent(event.HierarchyEvent arg0)`
- protected void `processHierarchyEvent(event.HierarchyEvent arg0)`
- protected void `processInputMethodEvent(event.InputMethodEvent arg0)`
- protected void `processKeyEvent(event.KeyEvent arg0)`
- protected void `processMouseEvent(event.MouseEvent arg0)`
- protected void `processMouseMotionEvent(event.MouseEvent arg0)`
- protected void `processMouseWheelEvent(event.MouseWheelEvent arg0)`
- private void `readObject(java.io.ObjectInputStream arg0) throws java.lang.ClassNotFoundException, java.io.IOException`
- final void `relocateComponent()`
- public void `remove(MenuComponent arg0)`
- public synchronized void `removeComponentListener(event.ComponentListener arg0)`
- public synchronized void `removeFocusListener(event.FocusListener arg0)`
- public void `removeHierarchyBoundsListener(event.HierarchyBoundsListener arg0)`
- public void `removeHierarchyListener(event.HierarchyListener arg0)`
- public synchronized void `removeInputMethodListener(event.InputMethodListener arg0)`
- public synchronized void `removeKeyListener(event.KeyListener arg0)`
- public synchronized void `removeMouseListener(event.MouseListener arg0)`
- public synchronized void `removeMouseMotionListener(event.MouseMotionListener arg0)`

- `public synchronized void removeMouseListener(event.MouseWheelListener arg0)`
- `public void removeNotify()`
- `public void removePropertyChangeListener(java.beans.PropertyChangeListener arg0)`
- `public void removePropertyChangeListener(java.lang.String arg0, java.beans.PropertyChangeListener arg1)`
- `public void repaint()`
- `public void repaint(int arg0, int arg1, int arg2, int arg3)`
- `public void repaint(long arg0)`
- `public void repaint(long arg0, int arg1, int arg2, int arg3, int arg4)`
- `private void repaintParentIfNeeded(int arg0, int arg1, int arg2, int arg3)`
- `public void requestFocus()`
- `protected boolean requestFocus(boolean arg0)`
- `boolean requestFocus(boolean arg0, sun.awt.CausedFocusEvent.Cause arg1)`
- `boolean requestFocus(sun.awt.CausedFocusEvent.Cause arg0)`
- `private static requestFocusController`
- `final boolean requestFocusHelper(boolean arg0, boolean arg1)`
- `final boolean requestFocusHelper(boolean arg0, boolean arg1, sun.awt.CausedFocusEvent.Cause arg2)`
- `public boolean requestFocusInWindow()`
- `protected boolean requestFocusInWindow(boolean arg0)`
- `boolean requestFocusInWindow(boolean arg0, sun.awt.CausedFocusEvent.Cause arg1)`
- `boolean requestFocusInWindow(sun.awt.CausedFocusEvent.Cause arg0)`
- `public void reshape(int arg0, int arg1, int arg2, int arg3)`
- `private void reshapeNativePeer(int arg0, int arg1, int arg2, int arg3, int arg4)`
- `public void resize(Dimension arg0)`
- `public void resize(int arg0, int arg1)`
- `public void revalidate()`
- `public static final RIGHT_ALIGNMENT`
- `private static final serialVersionUID`
- `void setAutoFocusTransferOnDisposal(boolean arg0)`
- `public void setBackground(Color arg0)`
- `public void setBounds(int arg0, int arg1, int arg2, int arg3)`
- `public void setBounds(Rectangle arg0)`
- `void setBoundsOp(int arg0)`
- `public void setComponentOrientation(ComponentOrientation arg0)`
- `public void setCursor(Cursor arg0)`
- `public synchronized void setDropTarget(dnd.DropTarget arg0)`
- `public void setEnabled(boolean arg0)`
- `public void setFocusable(boolean arg0)`
- `final void setFocusTraversalKeys_NoIDCheck(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalKeys(int arg0, java.util.Set arg1)`
- `public void setFocusTraversalKeysEnabled(boolean arg0)`
- `public void setFont(Font arg0)`
- `public void setForeground(Color arg0)`
- `void setGraphicsConfiguration(GraphicsConfiguration arg0)`
- `public void setIgnoreRepaint(boolean arg0)`
- `public void setLocale(java.util.Locale arg0)`
- `public void setLocation(int arg0, int arg1)`
- `public void setLocation(Point arg0)`
- `public void setMaximumSize(Dimension arg0)`
- `public void setMinimumSize(Dimension arg0)`
- `public void setName(java.lang.String arg0)`



- `public void setPreferredSize(Dimension arg0)`
- `static synchronized void setRequestFocusController(sun.awt.RequestFocusController arg0)`
- `public void setSize(Dimension arg0)`
- `public void setSize(int arg0, int arg1)`
- `public void setVisible(boolean arg0)`
- `public void show()`
- `public void show(boolean arg0)`
- `public Dimension size()`
- `final void subtractAndApplyShape(sun.java2d.pipe.Region arg0)`
- `final void subtractAndApplyShapeBelowMe()`
- `static final textListenerK`
- `public static final TOP_ALIGNMENT`
- `public String toString()`
- `public void transferFocus()`
- `boolean transferFocus(boolean arg0)`
- `public void transferFocusBackward()`
- `boolean transferFocusBackward(boolean arg0)`
- `public void transferFocusUpCycle()`
- `public void update(Graphics arg0)`
- `final void updateCursorImmediately()`
- `boolean updateGraphicsData(GraphicsConfiguration arg0)`
- `void updateZOrder()`
- `private volatile valid`
- `public void validate()`
- `visible`
- `width`
- `transient windowClosingException`
- `static final windowFocusListenerK`
- `static final windowListenerK`
- `static final windowStateListenerK`
- `private void writeObject(java.io.ObjectOutputStream arg0) throws java.io.IOException`
- `x`
- `y`

## 5.7 Class WinDialog

Das Dialogfenster, in dem die Spieler ihre Namen eingeben muessen bevor das Spiel beginnt.

### 5.7.1 Declaration

```
public class WinDialog
    extends java.lang.Object
```

### 5.7.2 Field summary

- btnExit** Beenden Button im Dialog.
- btnRestart** Neustart Button im Dialog.
- dialog** Dialog.
- spieler** Der Spieler der gewonnen hat.
- view** Die View des Spiels.

### 5.7.3 Constructor summary

**WinDialog(Frame, View, Spieler)** Das Dialogfenster wird erstellt.

### 5.7.4 Method summary

**showWinDialog(Frame)** Zeigt den Dialog zur Eingabe der Spielernamen an.

### 5.7.5 Fields

- **private data.Spieler spieler**
  - Der Spieler der gewonnen hat.
- **private javax.swing.JButton btnRestart**
  - Neustart Button im Dialog.
- **private javax.swing.JButton btnExit**
  - Beenden Button im Dialog.
- **private javax.swing.JDialog dialog**
  - Dialog.
- **private View view**
  - Die View des Spiels.

### 5.7.6 Constructors

- **WinDialog**

```
public WinDialog(java.awt.Frame frame, View view, data.Spieler  
    spieler)
```

- **Description**  
Das Dialogfenster wird erstellt.
- **Parameters**
  - \* **frame** – Das Elternfenster zu dem der Dialog gehoeren soll.
  - \* **spieler** – Die Spieler.

### 5.7.7 Methods

- **showWinDialog**

```
private void showWinDialog(java.awt.Frame o)
```

- **Description**  
Zeigt den Dialog zur Eingabe der Spielernamen an.
- **Parameters**
  - \* **o** – Das Elternfenster zu dem der Dialog gehoeren soll.

## 5.8 Class WinDialog.EnterKeyListener

Ein KeyListener fuer die Enter-Taste.

### 5.8.1 Declaration

```
class WinDialog.EnterKeyListener
extends java.awt.event.KeyAdapter
```

### 5.8.2 Constructor summary

**WinDialog.EnterKeyListener()**

### 5.8.3 Method summary

**keyPressed(KeyEvent)**

### 5.8.4 Constructors

- **WinDialog.EnterKeyListener**

`WinDialog.EnterKeyListener()`

### 5.8.5 Methods

- **keyPressed**

`void keyPressed(java.awt.event.KeyEvent arg0)`

### 5.8.6 Members inherited from class KeyAdapter

`java.awt.event.KeyAdapter`

- `public void keyPressed(KeyEvent arg0)`
- `public void keyReleased(KeyEvent arg0)`
- `public void keyTyped(KeyEvent arg0)`