

sum.komponenten

Class Ausrichtung

[java.lang.Object](#)



All Implemented Interfaces:
[Serializable](#)

```
public class Ausrichtung
extends Object
implements Serializable
```

Konstanten fuer Ausrichtung des Texts in Etiketten und Textfeldern.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
static int	HORIZONTAL
static int	LINKS
static int	LINKSBUENDIG
static int	MITTE
static int	RECHTS
static int	RECHTSBUENDIG
static int	VERTIKAL
static int	ZENTRIERT

Constructor Summary
Ausrichtung()

Method Summary
Methods inherited from class java.lang. Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail
HORIZONTAL
public static final int HORIZONTAL
See Also: Constant Field Values

LINKS

public static final int LINKS

See Also:
[Constant Field Values](#)

LINKSBUENDIG

public static final int LINKSBUENDIG

See Also:
[Constant Field Values](#)

MITTE

public static final int MITTE

See Also:
[Constant Field Values](#)

RECHTS

public static final int RECHTS

See Also:
[Constant Field Values](#)

RECHTSBUENDIG

public static final int RECHTSBUENDIG

See Also:
[Constant Field Values](#)

VERTIKAL

public static final int VERTIKAL

See Also:
[Constant Field Values](#)

ZENTRIERT

public static final int ZENTRIERT

See Also:
[Constant Field Values](#)

Constructor Detail

Ausrichtung

public Ausrichtung()

sum.komponenten

Class Auswahl

[java.lang.Object](#)



All Implemented Interfaces:
[Serializable](#)

```
public class Auswahl
extends Komponente
implements Serializable
```

Eine Auswahl ist die Klasse fuer Java-Choices (Popup-Menus). Auswahlen reagieren auf einen Mausklick.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten.Komponente
hatComponent , kenntFenster

Constructor Summary

Auswahl (double pLinks, double pOben, double pBreite, double pHoehe) Die Auswahl wird erzeugt.
Auswahl (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe) Die Auswahl wird erzeugt.

Method Summary

protected void	bekommtFokus () Die Auswahl erhaelt den Fokus.
void	entferneAlleZeilen () Die momentane Anzahl der Zeilen in der Auswahl wird auf 0 gesetzt.
protected void	gewaehlt (int pIndex, String pText) Die Auswahl reagiert auf einen Mausklick.
void	haengeAn (String pText) Eine neue Zeile wird and die Auswahl angehaengt.
int	index () Der momentane Index der Auswahl wird abgefragt.
void	setzeBearbeiterGeaendert (String pBearbeiter) Die Methode zur Bearbeitung des Geaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
String	text () Der momentane Text der Auswahl wird abgefragt.
protected void	verLiertFokus () Die Auswahl verliert den Fokus.
void	waehle (int pIndex) Der momentane Index der Auswahl wird geaendert.
void	waehle (String pText) Der momentane Text der Auswahl wird geaendert.
int	zeilenAnzahl () Die momentane Anzahl der Zeilen in der Auswahl wird abgefragt.

Methods inherited from class <code>sum.komponenten.Komponente</code>
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class <code>java.lang.Object</code>
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

Auswahl

```
public Auswahl(double pLinks,
               double pOben,
               double pBreite,
               double pHoehe)
```

Die Auswahl wird erzeugt. Position sowie Breite und Hoehe werden als Parameter uebergeben. Die Auswahl befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente

Auswahl

```
public Auswahl(Fenster pFenster,
               double pLinks,
               double pOben,
               double pBreite,
               double pHoehe)
```

Die Auswahl wird erzeugt. Position sowie Breite und Hoehe werden als Parameter uebergeben. Die Auswahl befindet sich auf dem Fenster.

- Parameters:**
- pFenster - das Fenster, das die Komponente enthält
 - pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente

Method Detail

bekommtFokus

```
protected void bekommtFokus()
```

Die Auswahl erhaelt den Fokus.

entferneAlleZeilen

```
public void entferneAlleZeilen()
```

Die momentane Anzahl der Zeilen in der Auswahl wird auf 0 gesetzt.

gewaehlt

```
protected void gewaehlt(int pIndex,
                       String pText)
```

Die Auswahl reagiert auf einen Mausklick.

haengeAn

```
public void haengeAn(String pText)
```

Eine neue Zeile wird and die Auswahl angehaengt.

Parameters:
pText - der Text, der am Ende als neue Zeileangehaengt wird

index

```
public int index()
```

Der momentane Index der Auswahl wird abgefragt.

Returns:
der aktuelle Index der Auswahl

setzeBearbeiterGeaendert

```
public void setzeBearbeiterGeaendert(String pBearbeiter)
```

Die Methode zur Bearbeitung des Geaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Auswahl geaendert wurde.

text

```
public String text()
```

Der momentane Text der Auswahl wird abgefragt.

Returns:
der aktuelle text der Auswahl

verliertFokus

```
protected void verliertFokus()
```

Die Auswahl verliert den Fokus.

waehle

```
public void waehle(int pIndex)
```

Der momentane Index der Auswahl wird geaendert.

Parameters:
pIndex - der neue Index der Auswahl

waehle

```
public void waehle(String pText)
```

Der momentane Text der Auswahl wird geaendert.

Parameters:
pText - der neue Text der Auswahl

zeilenAnzahl

```
public int zeilenAnzahl()
```

Die momentane Anzahl der Zeilen in der Auswahl wird abgefragt.

Returns:
die Anzahl der Zeilen der Auswahl

sum.komponenten

Class Etikett



All Implemented Interfaces:
[Serializable](#)

```
public class Etikett
extends Textkomponente
implements Serializable
```

Ein Etikett ist die eingedeutschte Klasse fuer Java-Labels. Etiketten dienen dazu, Bereiche zu beschriften oder Informationen zu schreiben.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary

Etikett (double pLinks, double pOben, double pBreite, double pHoehe, double pZahl)	Das Etikett wird erzeugt und bekommt den Text pZahl.
Etikett (double pLinks, double pOben, double pBreite, double pHoehe, int pZahl)	Das Etikett wird erzeugt und bekommt den Text pZahl.
Etikett (double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Das Etikett wird erzeugt und bekommt den Text pText.
Etikett (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, double pZahl)	Das Etikett wird erzeugt und bekommt den Text pZahl.
Etikett (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, int pZahl)	Das Etikett wird erzeugt und bekommt den Text pZahl.
Etikett (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Das Etikett wird erzeugt und bekommt den Text pText.

Method Summary

void	fuegeEin (String pText, int pStelle)	Der Text pText wird an Position pStelle eingefuegt.
void	haengeAn (char pZeichen)	Das Zeichen pZeichen wird am Ende angehaengt.
void	haengeAn (double pZahl)	Das Kommazahl pZahl wird am Ende angehaengt.
void	haengeAn (int pZahl)	Das ganze Zahl pZahl wird am Ende angehaengt.
void	haengeAn (String pText)	

	Der Text pText wird am Ende angehaengt.
String	inhaltAlsText() Der Text des Etiketts wird als String zurueckgegeben.
void	setzeAusrichtung (int pAusrichtung) Die Ausrichtung des Texts im Etikett wird geaendert.
void	setzeInhalt (String pText) Das Etikett erhaelt einen neuen Inhalt.

Methods inherited from class sum.komponenten.Textkomponente
inhaltAlsGanzeZahl , inhaltAlsLangeGanzeZahl , inhaltAlsZahl , inhaltIstGanzeZahl , inhaltIstLangeGanzeZahl , inhaltIstText , inhaltIstZahl , init , schriftfarbe , schriftFarbe , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeSchriftart , setzeSchriftArt , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftFarbe , setzeSchriftgroesse , setzeSchriftGroesse , setzeSchriftstil , setzeSchriftStil

Methods inherited from class sum.komponenten.Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

Etikett

```
public Etikett(double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               double pZahl)
```

Das Etikett wird erzeugt und bekommt den Text pZahl. Position, Breite und Hoehe werden als Parameter uebergeben. Das Etikett befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pZahl - der Inhalt der Komponente

Etikett

```
public Etikett(double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               int pZahl)
```

Das Etikett wird erzeugt und bekommt den Text pZahl. Position, Breite und Hoehe werden als Parameter uebergeben. Das Etikett befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pZahl - der Inhalt der Komponente

Etikett

```
public Etikett(double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               String pText)
```

Das Etikett wird erzeugt und bekommt den Text pText. Position, Breite und Hoehe werden als Parameter

uebergeben. Das Etikett befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Etikett

```
public Etikett(Fenster pFenster,
               double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               double pZahl)
```

Das Etikett wird erzeugt und bekommt den Text pZahl. Position, Breite und Hoehe werden als Parameter uebergeben. Das Etikett befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pZahl - der Inhalt der Komponente

Etikett

```
public Etikett(Fenster pFenster,
               double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               int pZahl)
```

Das Etikett wird erzeugt und bekommt den Text pZahl. Position, Breite und Hoehe werden als Parameter uebergeben. Das Etikett befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pZahl - der Inhalt der Komponente

Etikett

```
public Etikett(Fenster pFenster,
               double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               String pText)
```

Das Etikett wird erzeugt und bekommt den Text pText. Position, Breite und Hoehe werden als Parameter uebergeben. Das Etikett befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Method Detail

fuegeEin

```
public void fuegeEin(String pText,  
                    int pStelle)
```

Der Text pText wird an Position pStelle eingefuegt.

Parameters:

pText - der Text, der eingefuegt wird

pStelle - die Stelle, wo der Text eingefuegt wird

haengeAn

```
public void haengeAn(char pZeichen)
```

Das Zeichen pZeichen wird am Ende angehaengt.

Parameters:

pZeichen - das Zeichen, das am Ende angehaengt wird

haengeAn

```
public void haengeAn(double pZahl)
```

Das Kommazahl pZahl wird am Ende angehaengt.

Parameters:

pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

```
public void haengeAn(int pZahl)
```

Das ganze Zahl pZahl wird am Ende angehaengt.

Parameters:

pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

```
public void haengeAn(String pText)
```

Der Text pText wird am Ende angehaengt.

Parameters:

pText - der Text, der am Ende angehaengt wird

inhaltAlsText

```
public String inhaltAlsText()
```

Der Text des Etiketts wird als String zurueckgegeben.

Specified by:

[inhaltAlsText](#) in class [Textkomponente](#)

Returns:

der Inhalt der Komponente als Zeichenkette (String)

setzeAusrichtung

```
public void setzeAusrichtung(int pAusrichtung)
```

Die Ausrichtung des Texts im Etikett wird geaendert.

Parameters:

pAusrichtung - die neue Ausrichtung des Textes (siehe Klasse Ausrichtung)

setzeInhalt

```
public void setzeInhalt(String pText)
```

Das Etikett erhaelt einen neuen Inhalt.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der Text, der als neuer Inhalt gesetzt wird

[Overview](#) [Package](#) **[Class](#)** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.komponenten

Class Fortschrittsbalken

[java.lang.Object](#)



All Implemented Interfaces:
[Serializable](#)

```
public class Fortschrittsbalken
extends Komponente
implements Serializable
```

Ein Fortschrittsbalken ist die Klasse fuer Java-Sliders. Rollbalken reagieren auf die Verschiebung des Reglers mit der Maus.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary

Fortschrittsbalken (double pLinks, double pOben, double pBreite, double pHoehe, int pMinWert, int pMaxWert)	Der Fortschrittsbalken wird erzeugt.
Fortschrittsbalken (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, int pMinWert, int pMaxWert)	Der Fortschrittsbalken wird erzeugt.
Fortschrittsbalken (Fenster pFenster, int pStil, int pMinWert, int pMaxWert)	Der Fortschrittsbalken wird erzeugt.
Fortschrittsbalken (int pStil, int pMinWert, int pMaxWert)	Der Fortschrittsbalken wird erzeugt.

Method Summary

int	maximum ()	Das momentane Maximum des Fortschrittsbalkens wird abgefragt.
int	minimum ()	Das momentane Minimum des Fortschrittsbalkens wird abgefragt.
void	setzeMaximum (int pWert)	Der Fortschrittsbalken erhaelt eine neues Maximum.
void	setzeMinimum (int pWert)	Der Fortschrittsbalken erhaelt eine neues Minimum.
void	setzeWert (int pWert)	Der Fortschrittsbalken erhaelt eine neuen aktuellen Wert.
int	wert ()	Die momentane Position (Wert) des Fortschrittsbalkens wird abgefragt.

Methods inherited from class sum.komponenten. Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennnen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Fortschrittsbalken

```
public Fortschrittsbalken(double pLinks,  
                          double pOben,  
                          double pBreite,  
                          double pHoehe,  
                          int pMinWert,  
                          int pMaxWert)
```

Der Fortschrittsbalken wird erzeugt. Der Minimalwert und Maximalwert sowie Position, Breite und Hoehe werden als Parameter uebergeben. Der Fortschrittsbalken befindet sich auf dem Bildschirm.

Parameters:

pLinks - der Abstand der Komponente vom linken Fensterrand
pOben - der Abstand der Komponente vom oberen Fensterrand
pBreite - die Breite der Komponente
pHoehe - die Hoehe der Komponente
pMinWert - der Minimalwert
pMaxWert - der Maximalwert

Fortschrittsbalken

```
public Fortschrittsbalken(Fenster pFenster,  
                          double pLinks,  
                          double pOben,  
                          double pBreite,  
                          double pHoehe,  
                          int pMinWert,  
                          int pMaxWert)
```

Der Fortschrittsbalken wird erzeugt. Der Minimalwert und Maximalwert sowie Position, Breite und Hoehe werden als Parameter uebergeben. Der Fortschrittsbalken befindet sich auf dem Fenster.

Parameters:

pFenster - das Fenster, das die Komponente enthält
pLinks - der Abstand der Komponente vom linken Fensterrand
pOben - der Abstand der Komponente vom oberen Fensterrand
pBreite - die Breite der Komponente
pHoehe - die Hoehe der Komponente
pMinWert - der Minimalwert
pMaxWert - der Maximalwert

Fortschrittsbalken

```
public Fortschrittsbalken(Fenster pFenster,  
                          int pStil,  
                          int pMinWert,  
                          int pMaxWert)
```

Der Fortschrittsbalken wird erzeugt. Der Stil ist JProgressBar.VERTICAL oder JProgressBar.HORIZONTAL. Ausserdem werden Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe wird im Anwendungsprogramm festgelegt. Der Fortschrittsbalken befindet sich auf dem Fenster.

Parameters:

pFenster - das Fenster, das die Komponente enthält
pStil - die Ausrichtung des Fortschrittsbalken (siehe Klasse Ausrichtung)
pMinWert - der Minimalwert
pMaxWert - der Maximalwert

Fortschrittsbalken

```
public Fortschrittsbalken(int pStil,  
                          int pMinWert,  
                          int pMaxWert)
```

Der Fortschrittsbalken wird erzeugt. Der Stil ist JProgressBar.VERTICAL oder JProgressBar.HORIZONTAL. Ausserdem werden Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe wird im Anwendungsprogramm festgelegt.

- Parameters:**
- pStil - die Ausrichtung des Forschrittbalken (siehe Klasse Ausrichtung)
 - pMinWert - der Minimalwert
 - pMaxWert - der Maximalwert

Method Detail

maximum

```
public int maximum()
```

Das momentane Maximum des Fortschrittsbalkens wird abgefragt.

- Returns:**
- das aktuelle Maximum

minimum

```
public int minimum()
```

Das momentane Minimum des Fortschrittsbalkens wird abgefragt.

- Returns:**
- das aktuelle Minimum

setzeMaximum

```
public void setzeMaximum(int pWert)
```

Der Fortschrittsbalken erhaelt eine neues Maximum.

- Parameters:**
- pWert - das neue Maximum

setzeMinimum

```
public void setzeMinimum(int pWert)
```

Der Fortschrittsbalken erhaelt eine neues Minimum.

- Parameters:**
- pWert - das neue Minimum

setzeWert

```
public void setzeWert(int pWert)
```

Der Fortschrittsbalken erhaelt eine neuen aktuellen Wert.

- Parameters:**
- pWert - der neue Wert des Fortschrittbalkens

wert

```
public int wert()
```

Die momentane Position (Wert) des Fortschrittsbalkens wird abgefragt.

- Returns:**
- der aktuelle Wert des Fortschrittbalkens

sum.komponenten

Class Kennwortfeld



All Implemented Interfaces:
[Serializable](#)

```
public class Kennwortfeld
extends Textfeld
implements Serializable
```

Ein Kennwortfeld ist eine Unterklasse vom Textfeld. Eingaben im Kennwortfeld werden durch sogenannte Echozeichen dargestellt. Wenn etwas in ein Kennwortfeld getippt werden soll, muss es vorher mit einem Mausklick aktiviert werden, d.h. das Kennwortfeld muss den Fokus haben.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary	
protected JPasswordField	hatPasswordField
protected String	zEingabeBestaetigtBearbeiter

Fields inherited from class sum.komponenten. Textfeld
hatTextField

Fields inherited from class sum.komponenten. Markierungskomponente
zInhaltGeaendertBearbeiter , zMarkierungGeaendertBearbeiter

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary	
Kennwortfeld (double pLinks, double pOben, double pBreite, double pHoehe, char pEchozeichen)	Das Kennwortfeld wird erzeugt und enthaelt den uebergebenen Text.
Kennwortfeld (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, char pEchozeichen)	Das Kennwortfeld wird erzeugt und enthaelt den uebergebenen Text.

Method Summary	
protected void	bekommtFokus () Das Kennwortfeld erhaelt den Fokus.
protected void	eingabeBestaetigt ()

	Das Kennwortfeld reagiert auf die Eingabebestaetigung (Enter).
protected void	inhaltGeaendert() Das Kennwortfeld reagiert auf Veraenderungen.
char	kennwortzeichen() Das Echozeichen wird zurueckgegeben.
char	kennwortZeichen() Das Echozeichen wird zurueckgegeben.
protected void	markierungGeaendert() Das Kennwortfeld reagiert auf Veraenderungen der Markierung.
void	setzeBearbeiterEingabeBestaetigt() (String pBearbeiter) Die Methode zur Bearbeitung des EingabeBestaetigt-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeKennwortzeichen() (char pZeichen) Das Echozeichen wird festgelegt.
void	setzeKennwortZeichen() (char pZeichen) Das Echozeichen wird festgelegt.
protected void	verliertFokus() Das Kennwortfeld verliert den Fokus.

Methods inherited from class sum.komponenten. Textfeld
fuegeEin , haengeAn , haengeAn , haengeAn , haengeAn , inhaltAlsText , istMarkiert , loesche , loescheAlles , loescheMarkierung , markiereAlles , markiereNichts , markierterInhalt , markierungsAnfang , markierungsEnde , setzeAusrichtung , setzeInhalt , setzeMarkierung , teihinhalt

Methods inherited from class sum.komponenten. Markierungskomponente
setzeBearbeiterInhaltGeaendert , setzeBearbeiterMarkierungGeaendert , teilInhalt

Methods inherited from class sum.komponenten. Textkomponente
inhaltAlsGanzeZahl , inhaltAlsLangeGanzeZahl , inhaltAlsZahl , inhaltIstGanzeZahl , inhaltIstLangeGanzeZahl , inhaltIstText , inhaltIstZahl , init , schriftfarbe , schriftFarbe , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeSchriftart , setzeSchriftArt , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftgroesse , setzeSchriftGroesse , setzeSchriftstil , setzeSchriftStil

Methods inherited from class sum.komponenten. Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang. Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

hatPasswordField

protected [JPasswordField](#) hatPasswordField

zEingabeBestaetigtBearbeiter

protected [String](#) zEingabeBestaetigtBearbeiter

Constructor Detail

Kennwortfeld

```
public Kennwortfeld(double pLinks,
                    double pOben,
                    double pBreite,
                    double pHoehe,
                    char pEchozeichen)
```

Das Kennwortfeld wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Das Kennwortfeld befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pEchozeichen - das Zeichen, das an Stelle des eingegebenen Textes dargestellt wird

Kennwortfeld

```
public Kennwortfeld(Fenster pFenster,  
    double pLinks,  
    double pOben,  
    double pBreite,  
    double pHoehe,  
    char pEchozeichen)
```

Das Kennwortfeld wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Das Kennwortfeld befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthaelt
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pEchozeichen - das Zeichen, das an Stelle des eingegebenen Textes dargestellt wird

Method Detail

bekommtFokus

```
protected void bekommtFokus()
```

Das Kennwortfeld erhaelt den Fokus.

Overrides:

[bekommtFokus](#) in class [Textfeld](#)

eingabeBestaetigt

```
protected void eingabeBestaetigt()
```

Das Kennwortfeld reagiert auf die Eingabebestaetigung (Enter).

Overrides:

[eingabeBestaetigt](#) in class [Textfeld](#)

inhaltGeaendert

```
protected void inhaltGeaendert()
```

Das Kennwortfeld reagiert auf Veraenderungen.

Overrides:

[inhaltGeaendert](#) in class [Textfeld](#)

kennwortzeichen

```
public char kennwortzeichen()
```

Das Echozeichen wird zurueckgegeben.

Returns:

das aktuelle Echozeichen

kennwortZeichen

```
public char kennwortZeichen()
```

Das Echozeichen wird zurueckgegeben.

Returns:
das aktuelle Echozeichen

markierungGeaendert

protected void **markierungGeaendert**()

Das Kennwortfeld reagiert auf Veraenderungen der Markierung.

Overrides:
[markierungGeaendert](#) in class [Textfeld](#)

setzeBearbeiterEingabeBestaetigt

public void **setzeBearbeiterEingabeBestaetigt**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des EingabeBestaetigt-Ereignisses in der Ereignisanwendung wird festgelegt.

Overrides:
[setzeBearbeiterEingabeBestaetigt](#) in class [Textfeld](#)

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuManwendung, der aufgerufen wird, wenn in der Komponente Enter gedrueckt wurde.

setzeKennwortzeichen

public void **setzeKennwortzeichen**(char pZeichen)

Das Echozeichen wird festgelegt.

Parameters:
pZeichen - das neue Echozeichen

setzeKennwortZeichen

public void **setzeKennwortZeichen**(char pZeichen)

Das Echozeichen wird festgelegt.

Parameters:
pZeichen - das neue Echozeichen

verliertFokus

protected void **verliertFokus**()

Das Kennwortfeld verliert den Fokus.

Overrides:
[verliertFokus](#) in class [Textfeld](#)

sum.komponenten

Class Knopf



All Implemented Interfaces:
[Serializable](#)

Direct Known Subclasses:
[Radioknopf](#)

```
public class Knopf
extends Textkomponente
implements Serializable
```

Ein Knopf ist die Klasse fuer Java-Buttons. Knoepfe reagieren auf einen Klick mit der Maus. Knoepfe haben im Konstruktor als letzten Parameter eine Zeichenkette, die den Namen der Aktions-Methode enthaelt.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary	
protected AbstractButton	hatButton

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary	
Knopf ()	Dummy-Konstruktor fuer die Unterklasse RadioKnopf.
Knopf (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)	Der Knopf wird erzeugt und bekommt den Text pAufschrift.
Knopf (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift, String pGeklicktBearbeiter)	Der Knopf wird erzeugt und bekommt den Text pAufschrift.
Knopf (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)	Der Knopf wird erzeugt und bekommt den Text pAufschrift.
Knopf (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift, String pGeklicktBearbeiter)	Der Knopf wird erzeugt und bekommt den Text pAufschrift.

Method Summary	
protected void	bekommtFokus () Der Knopf erhaelt den Fokus.
String	inhaltAlsText () Die Beschriftung des Knopfs wird als String zurueckgegeben.
protected void	knopfGeklickt ()

	Der Knopf reagiert auf einen Mausklick, indem er die beim Konstruktor als Parameter uebergebene Methode der Anwendung aufruft.
void	setzeBearbeiterGeklickt (String pBearbeiter) Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeInhalt (String pText) Der Knopf erhaelt eine neue Aufschrift.
protected void	verliertFokus () Der Knopf verliert den Fokus.

Methods inherited from class sum.komponenten.Textkomponente
inhaltAlsGanzeZahl , inhaltAlsLangeGanzeZahl , inhaltAlsZahl , inhaltIstGanzeZahl , inhaltIstLangeGanzeZahl , inhaltIstText , inhaltIstZahl , init , schriftfarbe , schriftFarbe , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeSchriftart , setzeSchriftArt , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftFarbe , setzeSchriftgroesse , setzeSchriftGroesse , setzeSchriftstil , setzeSchriftStil

Methods inherited from class sum.komponenten.Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

hatButton

protected [AbstractButton](#) **hatButton**

Constructor Detail

Knopf

public **Knopf**()

Dummy-Konstruktor für die Unterklasse RadioKnopf.

Knopf

```
public Knopf(double pLinks,
             double pOben,
             double pBreite,
             double pHoehe,
             String pAufschrift)
```

Der Knopf wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der Name der Aktionsmethodee, die beim Druucken des Knopfes aufgerufen wird, muss noch festgelegt werden. Der Knopf befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Inhalt der Komponente

Knopf

```
public Knopf(double pLinks,
             double pOben,
             double pBreite,
             double pHoehe,
             String pAufschrift,
             String pGeklicktBearbeiter)
```

Der Knopf wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter

uebergeben. Der letzte Parameter ist der Name der Aktionsmethodee, die beim Druecken des Knopfes aufgerufen wird. Der Knopf befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Inhalt der Komponente
- pGeklicktBearbeiter - der Bezeichner des Dienstes, der aufgerufen wird, wenn der Knopf geklickt wird

Knopf

```
public Knopf(Fenster pFenster,
             double pLinks,
             double pOben,
             double pBreite,
             double pHoehe,
             String pAufschrift)
```

Der Knopf wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der Name der Aktionsmethodee, die beim Druecken des Knopfes aufgerufen wird, muss noch festgelegt werden. Der Knopf befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Inhalt der Komponente

Knopf

```
public Knopf(Fenster pFenster,
             double pLinks,
             double pOben,
             double pBreite,
             double pHoehe,
             String pAufschrift,
             String pGeklicktBearbeiter)
```

Der Knopf wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der letzte Parameter ist der Name der Aktionsmethodee, die beim Druecken des Knopfes aufgerufen wird. Der Knopf befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Inhalt der Komponente
- pGeklicktBearbeiter - der Bezeichner des Dienstes, der aufgerufen wird, wenn der Knopf geklickt wird

Method Detail

bekommtFokus

```
protected void bekommtFokus()

    Der Knopf erhaelt den Fokus.
```

inhaltAlsText

```
public String inhaltAlsText()
```

Die Beschriftung des Knopfs wird als String zurueckgegeben.

Specified by:

[inhaltAlsText](#) in class [Textkomponente](#)

Returns:

knopfGeklickt

protected void **knopfGeklickt**()

Der Knopf reagiert auf einen Mausklick, indem er die beim Konstruktor als Parameter uebergebene Methode der Anwendung aufruft.

setzeBearbeiterGeklickt

public void **setzeBearbeiterGeklickt**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Knopf geklickt wird

setzeInhalt

public void **setzeInhalt**([String](#) pText)

Der Knopf erhaelt eine neue Aufschrift.

Specified by:
[setzeInhalt](#) in class [Textkomponente](#)

Parameters:
pText - der Text, der als neuer Inhalt gesetzt wird

verliertFokus

protected void **verliertFokus**()

Der Knopf verliert den Fokus.

sum.komponenten

Class Komponente

[java.lang.Object](#)



Direct Known Subclasses:

[Auswahl](#),
[Fortschrittsbalken](#),
[Regler](#),
[Tabelle](#),
[Textkomponente](#)

```
public abstract class Komponente
extends Object
```

Die Klasse Komponente ist eine abstrakte Oberklasse fuer alle Komponenten der SuM-Komponentenbibliothek. Jede einer solchen Komponente hat eine Position, die durch die linke obere Ecke der Komponente angegeben ist. Neben der Groesse kann auch die Sichtbarkeit und die Aktivitaet gesteuert werden. Mit Hilfe der Tabulatortaste und der Maus kann der Fokus auf die jeweilige Komponente gesetzt werden.

Version:
 7.3 vom 11.10.2011

Author:
 Horst Hildebrecht, Bernard Schriek

Field Summary	
protected JComponent	hatComponent
protected Bildschirm	kenntFenster

Constructor Summary	
Komponente	()

Method Summary	
void	aktiviere () Die Komponente wird aktiviert.
boolean	besitztFokus () Liefert die Information, ob die Komponente den Fokus besitzt.
int	breite () Die Breite der Komponente wird zurueckgegeben.
void	deaktiviere () Die Komponente wird deaktiviert und reagiert nicht mehr.
Color	farbe () Die Hintergrundfarbe der Komponente wird zurueckgegeben.
protected String	fokusErhaltenBearbeiter () Der Name der Methode zur Bearbeitung des FokusErhalten-Ereignisses in der Ereignisanwendung wird geliefert.
protected String	fokusVerlorenBearbeiter () Der Name der Methode zur Bearbeitung des FokusVerloren-Ereignisses in der Ereignisanwendung wird geliefert.
void	gibFrei () aus Kompatibilitaetsgruenden
int	hoehe () Die Hoehe der Komponente wird zurueckgegeben.
protected void	init (double pLinks, double pOben, double pBreite, double pHoehe) Position, Breite und Hoehe der Komponente werden festgelegt.
boolean	istAktiv () Es wird zurueckgegeben, ob die Komponente aktiviert ist.
boolean	

	istSichtbar() Es wird zurueckgegeben, ob die Komponente sichtbar ist.
protected void	lerneKomponenteKennen() (Bildschirm pFenster, JComponent pKomponente) Die Komponente kennt die eigentliche Component aus der Java-Bibliothek.
int	links() Der linke Rand der Komponente wird zurueckgegeben.
int	oben() Der obere Rand der Komponente wird zurueckgegeben.
void	setzeBearbeiterFokusErhalten() (String pBearbeiter) Die Methode zur Bearbeitung des FokusErhalten-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeBearbeiterFokusVerloren() (String pBearbeiter) Die Methode zur Bearbeitung des FokusVerloren-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeFarbe() (Color pFarbe) Die Hintergrundfarbe der Komponente wird geaendert.
void	setzeFarbe() (int pFarbe) Die Hintergrundfarbe der Komponente wird veraendert
void	setzeFokus() Die Komponente erhaelt den Fokus.
protected void	setzeFokusWert() (boolean pFokus) Die Komponente merkt sich, ob sie den Fokus hat.
void	setzeGroesse() (double pBreite, double pHoehe) Die Komponente erhaelt eine neue Groesse.
void	setzeHinweis() (String pText) Die Komponente erhaelt einen Hinweis (Tooltip, Hint), der angezeigt wird, wenn sich die Maus ueber der Komponente befindet.
void	setzePosition() (double pWohinH, double pWohinV) Die Komponente erhaelt eine neue Position (Ecke oben links).
void	verstecke() Die Komponente wird unsichtbar, existiert aber weiter.
void	zeige() Die Komponente wird angezeigt.

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

hatComponent

protected [JComponent](#) hatComponent

kenntFenster

protected [Bildschirm](#) kenntFenster

Constructor Detail

Komponente

public **Komponente()**

Method Detail

aktiviere

public void **aktiviere()**

Die Komponente wird aktiviert.

besitztFokus


```
public boolean besitztFokus()
```

Liefert die Information, ob die Komponente den Fokus besitzt.

Returns:
true, wenn die Komponente den Fokus besitzt

breite

```
public int breite()
```

Die Breite der Komponente wird zurueckgegeben.

Returns:
die Breite der Komponente

deaktiviere

```
public void deaktiviere()
```

Die Komponente wird deaktiviert und reagiert nicht mehr.

farbe

```
public Color farbe()
```

Die Hintergrundfarbe der Komponente wird zurueckgegeben.

Returns:
die Hintergrundfarbe der Komponente

fokusErhaltenBearbeiter

```
protected String fokusErhaltenBearbeiter()
```

Der Name der Methode zur Bearbeitung des FokusErhalten-Ereignisses in der Ereignisanwendung wird geliefert.

fokusVerlorenBearbeiter

```
protected String fokusVerlorenBearbeiter()
```

Der Name der Methode zur Bearbeitung des FokusVerloren-Ereignisses in der Ereignisanwendung wird geliefert.

gibFrei

```
public void gibFrei()
```

aus Kompatibilitaetsgruenden

hoehe

```
public int hoehe()
```

Die Hoehe der Komponente wird zurueckgegeben.

Returns:
die Hoehe der Komponente

init

```
protected void init(double pLinks,  
                    double pOben,  
                    double pBreite,  
                    double pHoehe)
```

Position, Breite und Hoehe der Komponente werden festgelegt.

istAktiv

public boolean **istAktiv**()

Es wird zurueckgegeben, ob die Komponente aktiviert ist.

Returns:
true, wenn die Komponente aktiv ist

istSichtbar

public boolean **istSichtbar**()

Es wird zurueckgegeben, ob die Komponente sichtbar ist.

Returns:
true, wenn die Komponente sichtbar ist

lerneKomponenteKennen

protected void **lerneKomponenteKennen**([Bildschirm](#) pFenster,
[JComponent](#) pKomponente)

Die Komponente kennt die eigentliche Component aus der Java-Bibliothek.

links

public int **links**()

Der linke Rand der Komponente wird zurueckgegeben.

Returns:
der Abstand der Komponente vom linken Bildschirmrand

oben

public int **oben**()

Der obere Rand der Komponente wird zurueckgegeben.

Returns:
der Abstand der Komponente vom oberen Bildschirmrand

setzeBearbeiterFokusErhalten

public void **setzeBearbeiterFokusErhalten**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des FokusErhalten-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Komponente den Fokus erhalten hat.

setzeBearbeiterFokusVerloren

public void **setzeBearbeiterFokusVerloren**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des FokusVerloren-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Komponente den Fokus verloren hat.

setzeFarbe

public void **setzeFarbe**([Color](#) pFarbe)

Die Hintergrundfarbe der Komponente wird geaendert.

Parameters:

pFarbe - die neue Hintergrundfarbe der Komponente

setzeFarbe

```
public void setzeFarbe(int pFarbe)
```

Die Hintergrundfarbe der Komponente wird veraendert

Parameters:

pFarbe - die neue Hintergrundfarbe der Komponente

setzeFokus

```
public void setzeFokus()
```

Die Komponente erhaelt den Fokus.

setzeFokusWert

```
protected void setzeFokusWert(boolean pFokus)
```

Die Komponente merkt sich, ob sie den Fokus hat.

setzeGroesse

```
public void setzeGroesse(double pBreite,  
                        double pHoehe)
```

Die Komponente erhaelt eine neue Groesse.

Parameters:

pBreite - die neue Breite der Komponente

pHoehe - die neue Hoehe der Komponente

setzeHinweis

```
public void setzeHinweis(String pText)
```

Die Komponente erhaelt einen Hinweis (Tooltip, Hint), der angezeigt wird, wenn sich die Maus ueber der Komponente befindet.

Parameters:

pText - der Text des Tooltips

setzePosition

```
public void setzePosition(double pWohinH,  
                        double pWohinV)
```

Die Komponente erhaelt eine neue Position (Ecke oben links).

Parameters:

pWohinH - die neue horizontale Position der linken oberen Ecke

pWohinV - die neue vertikale Position der linken oberen Ecke

verstecke

```
public void verstecke()
```

Die Komponente wird unsichtbar, existiert aber weiter.

zeige

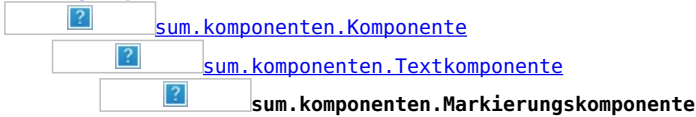
```
public void zeige()
```

Die Komponente wird angezeigt.

sum.komponenten

Class Markierungskomponente

java.lang.Object



Direct Known Subclasses:

[Textbereich](#), [Textfeld](#)

```
public abstract class Markierungskomponente
extends Textkomponente
```

Die Klasse Markierungskomponente ist eine abstrakte Oberklasse fuer die textorientierten Komponenten der SuM-Komponentenbibliothek, in denen der Benutzer einen Teil des Textes markieren kann. Neben den Diensten und Eigenschaften, die von der Oberklasse geerbt werden, besitzt eine Markierungskomponente Dienste zum Abfragen und Aendern der Markierung, zum Einfuegen, Anhaengen und Loeschen von Text

Version:

7.3 vom 11.10.2011

Author:

Horst Hildebrecht, Bernard Schriek

Field Summary

protected String	zInhaltGeaendertBearbeiter
protected String	zMarkierungGeaendertBearbeiter

Fields inherited from class sum.komponenten.[Textkomponente](#)

[zAktuellFont](#), [zSchriftArt](#), [zSchriftGroesse](#), [zSchriftStil](#)

Fields inherited from class sum.komponenten.[Komponente](#)

[hatComponent](#), [kenntFenster](#)

Constructor Summary

Markierungskomponente ()
--

Method Summary

abstract void	fuegeEin (String pText, int pStelle) Der Text pText wird an Position pStelle eingefuegt.
abstract void	haengeAn (char pZeichen) Das Zeichen pZeichen wird am Ende angehaengt.
abstract void	haengeAn (double pZahl) Die Kommazahl pZahl wird am Ende angehaengt.
abstract void	haengeAn (int pZahl) Das ganze Zahl pZahl wird am Ende angehaengt.
abstract void	haengeAn (String pText) Der Text pText wird am Ende angehaengt.
abstract boolean	istMarkiert () Es wird zurueckgegeben, ob der Text markiert ist.
abstract void	loesche (int pAnfang, int pEnde) Der durch die Parameter bestimmte Text wird geloescht.
void	loescheAlles () Der gesamte Text wird geloescht.

abstract void	loescheMarkierung() Der markierte Text wird geloescht.
abstract void	markiereAlles() Der gesamte Text wird markiert.
abstract void	markiereNichts() Der gesamte Text wird nicht markiert.
abstract String	markierterInhalt() Der im Textfeld markierte Text wird geliefert.
protected abstract void	markierungGeaendert() Die Markierungskomponente reagiert auf Veraenderungen der Markierung.
abstract int	markierungsAnfang() Es wird zurueckgegeben, von wo an der Text markiert ist.
abstract int	markierungsEnde() Es wird zurueckgegeben, bis wohin der Text markiert ist.
void	setzeBearbeiterInhaltGeaendert(String pBearbeiter) Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeBearbeiterMarkierungGeaendert(String pBearbeiter) Die Methode zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
abstract void	setzeMarkierung(int pAnfang, int pEnde) Der Text zwischen zwischen pAnfang und pEnde wird markiert.
abstract String	teillinhalt(int pAnfang, int pEnde) Der Inhalt der Textkomponente von pAnfang bis pEnde wird als String zurueckgegeben.
String	teilInhalt(int pAnfang, int pEnde) Der Inhalt der Markierungskomponente von pAnfang bis pEnde wird als String zurueckgegeben.

Methods inherited from class <code>sum.komponenten.Textkomponente</code>
inhaltAlsGanzeZahl , inhaltAlsLangeGanzeZahl , inhaltAlsText , inhaltAlsZahl , inhaltIstGanzeZahl , inhaltIstLangeGanzeZahl , inhaltIstText , inhaltIstZahl , init , schriftfarbe , schriftFarbe , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeSchriftart , setzeSchriftArt , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftgroesse , setzeSchriftGroesse , setzeSchriftstil , setzeSchriftStil

Methods inherited from class <code>sum.komponenten.Komponente</code>
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class <code>java.lang.Object</code>
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

zInhaltGeaendertBearbeiter

protected [String](#) zInhaltGeaendertBearbeiter

zMarkierungGeaendertBearbeiter

protected [String](#) zMarkierungGeaendertBearbeiter

Constructor Detail

Markierungskomponente

public `Markierungskomponente()`

Method Detail

fuegeEin

public abstract void **fuegeEin**([String](#) pText,

```
int pStelle)
```

Der Text pText wird an Position pStelle eingefuegt.

Parameters:

pText - der Text, der eingefuegt wird

pStelle - die Stelle, an der Text als neue Zeile eingefuegt wird

haengeAn

```
public abstract void haengeAn(char pZeichen)
```

Das Zeichen pZeichen wird am Ende angehaengt.

Parameters:

pZeichen - das Zeichen, das am Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(double pZahl)
```

Die Kommazahl pZahl wird am Ende angehaengt.

Parameters:

pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(int pZahl)
```

Das ganze Zahl pZahl wird am Ende angehaengt.

Parameters:

pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(String pText)
```

Der Text pText wird am Ende angehaengt.

Parameters:

pText - der Text, der am Ende angehaengt wird

istMarkiert

```
public abstract boolean istMarkiert()
```

Es wird zurueckgegeben, ob der Text markiert ist.

Returns:

true, wenn Text markiert ist

loesche

```
public abstract void loesche(int pAnfang,  
                             int pEnde)
```

Der durch die Parameter bestimmte Text wird geloescht.

Parameters:

pAnfang - erste Stelle

pEnde - letzte Stelle

loescheAlles

```
public void loescheAlles()
```

Der gesamte Text wird geloescht.

loescheMarkierung

```
public abstract void loescheMarkierung()

    Der markierte Text wird geloescht.
```

markiereAlles

```
public abstract void markiereAlles()

    Der gesamte Text wird markiert.
```

markiereNichts

```
public abstract void markiereNichts()

    Der gesamte Text wird nicht markiert.
```

markierterInhalt

```
public abstract String markierterInhalt()

    Der im Textfeld markierte Text wird geliefert.

Returns:
    der markierte Text
```

markierungGeaendert

```
protected abstract void markierungGeaendert()

    Die Markierungskomponente reagiert auf Veraenderungen der Markierung. Abstrakte Methode, muss in
    Unterklassen ueberschrieben werden.
```

markierungsAnfang

```
public abstract int markierungsAnfang()

    Es wird zurueckgegeben, von wo an der Text markiert ist.

Returns:
    die erste markierte Stelle
```

markierungsEnde

```
public abstract int markierungsEnde()

    Es wird zurueckgegeben, bis wohin der Text markiert ist.

Returns:
    die letzte markierte Stelle
```

setzeBearbeiterInhaltGeaendert

```
public void setzeBearbeiterInhaltGeaendert(String pBearbeiter)

    Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
    pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Inhalt der
    Komponente geaendert wurde.
```

setzeBearbeiterMarkierungGeaendert

```
public void setzeBearbeiterMarkierungGeaendert(String pBearbeiter)

    Die Methode zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
```


Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Markierung der Komponente geaendert wurde.

setzeMarkierung

```
public abstract void setzeMarkierung(int pAnfang,
                                     int pEnde)
```

Der Text zwischen zwischen pAnfang und pEnde wird markiert.

Parameters:
pAnfang - erste Stelle der Markierung
pEnde - letzte Stelle der Markierung

teillinhalt

```
public abstract String teillinhalt(int pAnfang,
                                   int pEnde)
```

Der Inhalt der Textkomponente von pAnfang bis pEnde wird als String zurueckgegeben.

teilInhalt

```
public String teilInhalt(int pAnfang,
                        int pEnde)
```

Der Inhalt der Markierungskomponente von pAnfang bis pEnde wird als String zurueckgegeben.

Parameters:
pAnfang - der Anfang des Textes
pEnde - das Ende des Textes

sum.komponenten

Class Menue

[java.lang.Object](#)



All Implemented Interfaces:
[Serializable](#)

```
public class Menue
extends Textkomponente
implements Serializable
```

Ein Menue ist die Klasse fuer Java-Menues (Fenster-Menues). Menues reagieren auf einen Mausklick.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary

	Menue (String pTitel) Das Menue mit dem angegebenen Titel wird im Menuebar erzeugt.
protected	Menue (String pTitel, JMenu pObermenu) Das Menue mit dem angegebenen Titel und Obermenue wird erzeugt.

Method Summary

void	aktiviere () Das Menue wird aktiviert.
void	aktiviereZeile (int pZeile) Eine Menuezeile wird aktiviert.
int	breite () Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.
void	deaktiviere () Das Menue wird deaktiviert und reagiert nicht mehr.
void	deaktiviereZeile (int pZeile) Eine Menuezeile wird deaktiviert und reagiert nicht mehr.
protected void	gewaehlt (String pAuftrag) Das Menue reagiert auf einen Mausklick.
void	haengeTrennungAn () Eine Trennzeile wird an das Menue angehaengt.
void	haengeZeileAn (String pText, char pZeichen, boolean pMitShift, String pAuftrag) Eine neue Zeile wird mit einer Beschleunigertaste an das Menue angehaengt.
void	haengeZeileAn (String pText, String pAuftrag) Eine neue Zeile wird an das Menue angehaengt.

int	hoehe() Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.
String	inhaltAlsText() Der Titel des Menues wird als String zurueckgegeben.
boolean	istAktiv() Es wird zurueckgegeben, ob das Menue aktiviert ist.
boolean	istSichtbar() Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert true.
boolean	istZeileAktiv(int pZeile) Es wird zurueckgegeben, ob eine Menuezeile aktiviert ist.
protected void	lerneKomponenteKennen(JComponent pKomponente)
int	links() Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.
Menue	neuesUntermenue(String pText) Ein Untermenue wird an das Menue angehaengt.
int	oben() Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.
void	setzeBearbeiterFokusErhalten(String pBearbeiter) Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeBearbeiterFokusVerloren(String pBearbeiter) Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeBearbeiterGewaeht(int pZeile, String pAuftrag) Die Methode zur Bearbeitung des Gewaeht-Ereignisses für eine Menuezeile in der Ereignisanwendung wird festgelegt.
void	setzeFarbe(Color pFarbe) Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeFokus() Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeGroesse(double pBreite, double pHoehe) Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeInhalt(String pText) Das Menue erhaelt einen neuen Titel.
void	setzePosition(double pWohinH, double pWohinV) Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	setzeSchriftArt(String pSchriftart) Die Schriftart des Menues wird veraendert
void	setzeSchriftFarbe(Color pFarbe) Die Schriftfarbe des Menues kann nicht veraendert werden.
void	setzeSchriftGroesse(int pGroesse) Die Schriftgroesse des Menues wird veraendert
void	setzeSchriftstil(int pStil) Der Schriftstil des Menues wird veraendert
void	setzeSchriftStil(int pStil) Der Schriftstil des Menues wird veraendert
void	verstecke() Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
void	zeige() Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.
int	zeilenAnzahl() Die momentane Anzahl der Zeilen des Menues wird abgefragt.

Methods inherited from class [sum.komponenten.Textkomponente](#)

[inhaltAlsGanzeZahl](#), [inhaltAlsLangeGanzeZahl](#), [inhaltAlsZahl](#), [inhaltIstGanzeZahl](#), [inhaltIstLangeGanzeZahl](#), [inhaltIstText](#), [inhaltIstZahl](#), [init](#), [schriftfarbe](#), [schriftFarbe](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeSchriftart](#), [setzeSchriftfarbe](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftgroesse](#)

Methods inherited from class [sum.komponenten.Komponente](#)

[besitztFokus](#), [farbe](#), [fokusErhaltenBearbeiter](#), [fokusVerlorenBearbeiter](#), [gibFrei](#), [init](#), [lerneKomponenteKennen](#), [setzeFarbe](#), [setzeFokusWert](#), [setzeHinweis](#)

Methods inherited from class [java.lang.Object](#)

Constructor Detail

Menue

```
public Menue(String pTitel)
```

Das Menue mit dem angegebenen Titel wird im Menuebar erzeugt.

Parameters:

pTitel - der Titel des Menues

Menue

```
protected Menue(String pTitel,  
                JMenu pObermenu)
```

Das Menue mit dem angegebenen Titel und Obermenue wird erzeugt. Ist kein Obermenue angegeben, wird es im Menuebar erzeugt.

Method Detail

aktiviere

```
public void aktiviere()
```

Das Menue wird aktiviert.

Overrides:

[aktiviere](#) in class [Komponente](#)

aktiviereZeile

```
public void aktiviereZeile(int pZeile)
```

Eine Menuezeile wird aktiviert.

Parameters:

pZeile - die aktivierte Zeile

breite

```
public int breite()
```

Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.

Overrides:

[breite](#) in class [Komponente](#)

Returns:

die Breite der Komponente

deaktiviere

```
public void deaktiviere()
```

Das Menue wird deaktiviert und reagiert nicht mehr.

Overrides:

[deaktiviere](#) in class [Komponente](#)

deaktiviereZeile

```
public void deaktiviereZeile(int pZeile)
```

Eine Menuezeile wird deaktiviert und reagiert nicht mehr.

Parameters:

pZeile - die deaktivierte Zeile

gewaehlt

```
protected void gewaehlt(String pAuftrag)
```

Das Menue reagiert auf einen Mausklick.

haengeTrennungAn

```
public void haengeTrennungAn()
```

Eine Trennzeile wird an das Menue angehaengt.

haengeZeileAn

```
public void haengeZeileAn(String pText,  
                        char pZeichen,  
                        boolean pMitShift,  
                        String pAuftrag)
```

Eine neue Zeile wird mit einer Beschleunigertaste an das Menue angehaengt.

Parameters:

pText - der Inhalt der Menuezeile

pZeichen - das Beschleunigerzeichen fuer die Menuezeile

pMitShift - bei true wird zwischen Gross- und Kleinschrift des Beschleunigerzeichens unterschieden

pAuftrag - der Name des Dienstes der SuMANwendung, der durch die Menuezeile aufgerufen wird

haengeZeileAn

```
public void haengeZeileAn(String pText,  
                        String pAuftrag)
```

Eine neue Zeile wird an das Menue angehaengt.

Parameters:

pText - der Inhalt der Menuezeile

pAuftrag - der Name des Dienstes der SuMANwendung, der durch die Menuezeile aufgerufen wird

hoehe

```
public int hoehe()
```

Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.

Overrides:

[hoehe](#) in class [Komponente](#)

Returns:

die Hoehe der Komponente

inhaltAlsText

```
public String inhaltAlsText()
```

Der Titel des Menues wird als String zurueckgegeben.

Specified by:

[inhaltAlsText](#) in class [Textkomponente](#)

Returns:

der Titel des Menues

istAktiv

```
public boolean istAktiv()
```

Es wird zurueckgegeben, ob das Menue aktiviert ist.

Overrides:

[istAktiv](#) in class [Komponente](#)

Returns:

true, wenn das Menue aktiviert ist

istSichtbar

```
public boolean istSichtbar()
```

Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert true.

Overrides:

[istSichtbar](#) in class [Komponente](#)

Returns:

true, wenn die Komponente sichtbar ist

istZeileAktiv

```
public boolean istZeileAktiv(int pZeile)
```

Es wird zurueckgegeben, ob eine Menuezeile aktiviert ist.

Parameters:

pZeile - die untersuchte Zeile

Returns:

true, wenn die Zeile aktiviert ist

lerneKomponenteKennen

```
protected void lerneKomponenteKennen(JComponent pKomponente)
```

links

```
public int links()
```

Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.

Overrides:

[links](#) in class [Komponente](#)

Returns:

der Abstand der Komponente vom linken Bildschirmrand

neuesUntermenue

```
public Menu neuesUntermenue(String pText)
```

Ein Untermenue wird an das Menue angehaengt.

Parameters:

pText - der Titel des Untermenues

Returns:

das neue Untermenue

oben

```
public int oben()
```

Die Anfrage kann fuer ein Menue nicht angewandt werden und liefert 0.

Overrides:

[oben](#) in class [Komponente](#)

Returns:

der Abstand der Komponente vom oberen Bildschirmrand

setzeBearbeiterFokusErhalten

public void **setzeBearbeiterFokusErhalten**([String](#) pBearbeiter)

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzeBearbeiterFokusErhalten](#) in class [Komponente](#)

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Komponente den Fokus erhalten hat.

setzeBearbeiterFokusVerloren

public void **setzeBearbeiterFokusVerloren**([String](#) pBearbeiter)

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzeBearbeiterFokusVerloren](#) in class [Komponente](#)

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Komponente den Fokus verloren hat.

setzeBearbeiterGewaeht

public void **setzeBearbeiterGewaeht**(int pZeile,
[String](#) pAuftrag)

Die Methode zur Bearbeitung des Gewaeht-Ereignisses für eine Menuezeile in der Ereignisanwendung wird festgelegt.

Parameters:

pZeile - die Zeile des Menues fuer die der Dienst festgelegt wird
pAuftrag - der Dienst, der mit der Menuezeile aufgerufen wird

setzeFarbe

public void **setzeFarbe**([Color](#) pFarbe)

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzeFarbe](#) in class [Komponente](#)

Parameters:

pFarbe - die neue Hintergrundfarbe der Komponente

setzeFokus

public void **setzeFokus**()

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzeFokus](#) in class [Komponente](#)

setzeGroesse

public void **setzeGroesse**(double pBreite,
double pHoehe)

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzeGroesse](#) in class [Komponente](#)

Parameters:

pBreite - die neue Breite der Komponente
pHoehe - die neue Hoehe der Komponente

setzeInhalt

```
public void setzeInhalt(String pText)
```

Das Menue erhaelt einen neuen Titel.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der neue Titel des Menues

setzePosition

```
public void setzePosition(double pWohinH,  
                          double pWohinV)
```

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:

[setzePosition](#) in class [Komponente](#)

Parameters:

pWohinH - die neue horizontale Position der linken oberen Ecke

pWohinV - die neue vertikale Position der linken oberen Ecke

setzeSchriftArt

```
public void setzeSchriftArt(String pSchriftart)
```

Die Schriftart des Menues wird veraendert

Overrides:

[setzeSchriftArt](#) in class [Textkomponente](#)

Parameters:

pSchriftart - die neue Schrift des Menues (siehe Klasse Schrift)

setzeSchriftFarbe

```
public void setzeSchriftFarbe(Color pFarbe)
```

Die Schriftfarbe des Menues kann nicht veraendert werden. Leerer Dienst.

Overrides:

[setzeSchriftFarbe](#) in class [Textkomponente](#)

Parameters:

pFarbe - die neue Schriftfarbe der Komponente

setzeSchriftGroesse

```
public void setzeSchriftGroesse(int pGroesse)
```

Die Schriftgroesse des Menues wird veraendert

Overrides:

[setzeSchriftGroesse](#) in class [Textkomponente](#)

Parameters:

pGroesse - die neue Groesse der Menueschrift

setzeSchriftstil

```
public void setzeSchriftstil(int pStil)
```

Der Schriftstil des Menues wird veraendert

Overrides:

[setzeSchriftstil](#) in class [Textkomponente](#)

Parameters:
pStil - der neue Schriftstil des Menues (siehe Klasse Schrift)

setzeSchriftStil

public void **setzeSchriftStil**(int pStil)

Der Schriftstil des Menues wird veraendert

Overrides:
[setzeSchriftStil](#) in class [Textkomponente](#)

Parameters:
pStil - der neue Schriftstil des Menues (siehe Klasse Schrift)

verstecke

public void **verstecke**()

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:
[verstecke](#) in class [Komponente](#)

zeige

public void **zeige**()

Die Methode kann fuer ein Menue nicht angewandt werden und ist daher leer.

Overrides:
[zeige](#) in class [Komponente](#)

zeilenAnzahl

public int **zeilenAnzahl**()

Die momentane Anzahl der Zeilen des Menues wird abgefragt.

Returns:
die Anzahl der Zeilen des Menues

sum.komponenten

Classes

Ausrichtung

Auswahl

Etikett

Fortschrittsbalken

Kennwortfeld

Knopf

Komponente

Markierungskomponente

Menue

Radiogruppe

Radioknopf

Regler

Schalter

Tabelle

Textbereich

Textfeld

Textkomponente

Zeichenbereich

Zeilenbereich

Package sum.komponenten

Class Summary	
Ausrichtung	Konstanten fuer Ausrichtung des Texts in Etiketten und Textfeldern.
Auswahl	Eine Auswahl ist die Klasse fuer Java-Choices (Popup-Menus).
Etikett	Ein Etikett ist die eingedeutschte Klasse fuer Java-Labels.
Fortschrittsbalken	Ein Fortschrittsbalken ist die Klasse fuer Java-Sliders.
Kennwortfeld	Ein Kennwortfeld ist eine Unterklasse vom Textfeld.
Knopf	Ein Knopf ist die Klasse fuer Java-Buttons.
Komponente	Die Klasse Komponente ist eine abstrakte Oberklasse fuer alle Komponenten der SuM-Komponentenbibliothek.
Markierungskomponente	Die Klasse Markierungskomponente ist eine abstrakte Oberklasse fuer die textorientierten Komponenten der SuM-Komponentenbibliothek, in denen der Benutzer einen Teil des Textes markieren kann.
Menue	Ein Menue ist die Klasse fuer Java-Menues (Fenster-Menues).
Radiogruppe	Implementierung der Swingklasse RadioGroup
Radioknopf	Ein Radioknopf ist eine Unterklasse vom Knopf.
Regler	Ein Regler ist die Klasse fuer Java-Sliders.
Schalter	Ein Schalter ist die Klasse fuer Java-Checkboxes.
Tabelle	Eine Tabelle ist die eingedeutschte Klasse fuer Java-JTables.
Textbereich	Ein Textbereich ist die Klasse fuer Java-TextAreas.
Textfeld	Ein Textfeld ist die Klasse fuer Java-TextFields.
Textkomponente	Die Klasse Textkomponente ist eine abstrakte Oberklasse fuer alle textorientierten Komponenten der SuM-Komponentenbibliothek.
Zeichenbereich	Ein Zeichenbereich ist die zeichenorientierte Unterklasse des Textbereichs.
Zeilenbereich	Ein Zeilenbereich ist die zeilenorientierte Unterklasse des Textbereichs.

Hierarchy For Package sum.komponenten

Package Hierarchies:

[All Packages](#)

Class Hierarchy

- java.lang.**Object**
 - sum.komponenten.**Ausrichtung** (implements java.io.[Serializable](#))
 - javax.swing.**ButtonGroup** (implements java.io.[Serializable](#))
 - sum.komponenten.**Radiogruppe** (implements java.io.[Serializable](#))
 - sum.komponenten.**Komponente**
 - sum.komponenten.**Auswahl** (implements java.io.[Serializable](#))
 - sum.komponenten.**Fortschrittsbalken** (implements java.io.[Serializable](#))
 - sum.komponenten.**Regler** (implements java.io.[Serializable](#))
 - sum.komponenten.**Tabelle** (implements javax.swing.[ScrollPaneConstants](#), java.io.[Serializable](#))
 - sum.komponenten.**Textkomponente**
 - sum.komponenten.**Etikett** (implements java.io.[Serializable](#))
 - sum.komponenten.**Knopf** (implements java.io.[Serializable](#))
 - sum.komponenten.**Radioknopf** (implements java.io.[Serializable](#))
 - sum.komponenten.**Markierungskomponente**
 - sum.komponenten.**Textbereich** (implements javax.swing.[ScrollPaneConstants](#), java.io.[Serializable](#))
 - sum.komponenten.**Zeichenbereich** (implements java.io.[Serializable](#))
 - sum.komponenten.**Zeilenbereich** (implements java.io.[Serializable](#))
 - sum.komponenten.**Textfeld** (implements java.io.[Serializable](#))
 - sum.komponenten.**Kennwortfeld** (implements java.io.[Serializable](#))
 - sum.komponenten.**Menue** (implements java.io.[Serializable](#))
 - sum.komponenten.**Schalter** (implements java.io.[Serializable](#))

sum.komponenten

Class Radiogruppe

[java.lang.Object](#)

 [javax.swing.ButtonGroup](#)

 **sum.komponenten.Radiogruppe**

All Implemented Interfaces:

[Serializable](#)

public class **Radiogruppe**

extends [ButtonGroup](#)

implements [Serializable](#)

Implementierung der Swingklasse RadioGroup

Version:

7.3 vom 11.10.2011

Author:

Bernard Schriek

See Also:

[Serialized Form](#)

Field Summary

Fields inherited from class javax.swing.[ButtonGroup](#)

[buttons](#)

Constructor Summary

[Radiogruppe](#)()

Method Summary

void [entferne](#)([Radioknopf](#) pKnopf)

void [fuegeEin](#)([Radioknopf](#) pKnopf)

Methods inherited from class javax.swing.[ButtonGroup](#)

[add](#), [clearSelection](#), [getButtonCount](#), [getElements](#), [getSelection](#), [isSelected](#), [remove](#), [setSelected](#)

Methods inherited from class java.lang.[Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Radiogruppe

public **Radiogruppe**()

Method Detail

entferne

public void **entferne**([Radioknopf](#) pKnopf)

fuegeEin

public void **fuegeEin**([Radioknopf](#) pKnopf)

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

sum.komponenten

Class Radioknopf



All Implemented Interfaces:

[Serializable](#)

```
public class Radioknopf
extends Knopf
implements Serializable
```

Ein Radioknopf ist eine Unterklasse vom Knopf. Radioknöpfe werden in einer Radiogruppe zusammengefasst. In der Gruppe kann nur ein Radioknopf aktiv sein.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary

protected JRadioButton	hatRadioButton
protected String	zGeklicktBearbeiter

Fields inherited from class sum.komponenten.Knopf

hatButton

Fields inherited from class sum.komponenten.Textkomponente

zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten.Komponente

hatComponent , kenntFenster

Constructor Summary

Radioknopf (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)	Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text.
Radioknopf (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift, String pGeklicktBearbeiter)	Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text.
Radioknopf (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)	Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text.
Radioknopf (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift, String pGeklicktBearbeiter)	Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text.

Method Summary

protected void	bekommtFokus ()	Der Knopf erhaelt den Fokus.
boolean	istGewaehlt ()	ist der Radioknopf ausgewaehlt.

Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Radioknopf befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Text, der neben der Komponente dargestellt wird
- pGeklicktBearbeiter - der Bezeichner des Dienstes, der aufgerufen wird, wenn der Knopf geklickt wird

Radioknopf

```
public Radioknopf(Fenster pFenster,
                  double pLinks,
                  double pOben,
                  double pBreite,
                  double pHoehe,
                  String pAufschrift)
```

Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Radioknopf befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthaelt
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Text, der neben der Komponente dargestellt wird

Radioknopf

```
public Radioknopf(Fenster pFenster,
                  double pLinks,
                  double pOben,
                  double pBreite,
                  double pHoehe,
                  String pAufschrift,
                  String pGeklicktBearbeiter)
```

Der Radioknopf wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Radioknopf befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthaelt
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pAufschrift - der Text, der neben der Komponente dargestellt wird
- pGeklicktBearbeiter - der Bezeichner des Dienstes, der aufgerufen wird, wenn der Knopf geklickt wird

Method Detail

bekommtFokus

```
protected void bekommtFokus()
```

Der Knopf erhaelt den Fokus.

Overrides:

[bekommtFokus](#) in class [Knopf](#)

istGewaeht

```
public boolean istGewaeht()
```

ist der Radioknopf ausgewaeht.

Returns:

true, wenn der Radioknopf gewaeht ist.

knopfGeklickt

protected void **knopfGeklickt**()

Der Knopf reagiert auf einen Mausklick, indem er die beim Konstruktor als Parameter uebergebene Methode der Anwendung aufruft.

Overrides:
[knopfGeklickt](#) in class [Knopf](#)

setzeBearbeiterGeklickt

public void **setzeBearbeiterGeklickt**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des Geklickt-Ereignisses in der Ereignisanwendung wird festgelegt.

Overrides:
[setzeBearbeiterGeklickt](#) in class [Knopf](#)

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Knopf geklickt wird

verliertFokus

protected void **verliertFokus**()

Der Knopf verliert den Fokus.

Overrides:
[verliertFokus](#) in class [Knopf](#)

waehle

public void **waehle**()

der Radioknopf wird ausgewaehlt.

sum.komponenten

Class Regler

[java.lang.Object](#)



All Implemented Interfaces:

[Serializable](#)

```
public class Regler
extends Komponente
implements Serializable
```

Ein Regler ist die Klasse fuer Java-Sliders. Regler reagieren auf die Verschiebung des Reglers mit der Maus.

Version:

7.3 vom 11.10.2011

Author:

Bernard Schriek, Horst Hildebrecht

See Also:

[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary

Regler (double pLinks, double pOben, double pBreite, double pHoehe, int pAnfangswert, int pMinwert, int pMaxwert)
Der Regler wird erzeugt.
Regler (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, int pAnfangswert, int pMinwert, int pMaxwert)
Der Regler wird erzeugt.
Regler (Fenster pFenster, int pStil, int pAnfangswert, int pMinwert, int pMaxwert)
Der Regler wird erzeugt.
Regler (int pStil, int pAnfangswert, int pMinwert, int pMaxwert)
Der Regler wird erzeugt.

Method Summary

protected void	bekommtFokus ()	Der Regler erhaelt den Fokus.
protected void	geaendert ()	Der Regler reagiert auf eine Bewegung.
int	maximum ()	Das momentane Maximum des Reglers wird abgefragt.
int	minimum ()	Das momentane Minimum des Reglers wird abgefragt.
void	setzeBearbeiterGeaendert (String pBearbeiter)	Die Methode zur Bearbeitung des Geaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeMaximum (int pWert)	Der Regler erhaelt eine neues Maximum.
void	setzeMinimum (int pWert)	Der Regler erhaelt eine neues Minimum.
void	setzeWert (int pWert)	Der Regler erhaelt eine neuen aktuellen Wert.
protected	verliertFokus ()	

void	Der Regler verliert den Fokus.
int	wert() Die momentane Position (Wert) des Reglers wird abgefragt.

Methods inherited from class sum.komponenten.Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

Regler

```
public Regler(double pLinks,  
              double pOben,  
              double pBreite,  
              double pHoehe,  
              int pAnfangswert,  
              int pMinwert,  
              int pMaxwert)
```

Der Regler wird erzeugt. Der Stil ist Ausrichtung.VERTICAL oder Ausrichtung.HORIZONTAL. Ausserdem werden Anfangswert, Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe werden als Parameter uebergeben.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pAnfangswert - der Anfangswert des Reglers
 - pMinwert - der Minimalwert des Reglers
 - pMaxwert - der Maximalwert des Reglers

Regler

```
public Regler(Fenster pFenster,  
              double pLinks,  
              double pOben,  
              double pBreite,  
              double pHoehe,  
              int pAnfangswert,  
              int pMinwert,  
              int pMaxwert)
```

Der Regler wird erzeugt. Der Stil ist Ausrichtung.VERTICAL oder Ausrichtung.HORIZONTAL. Ausserdem werden Anfangswert, Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe werden als Parameter uebergeben. Der Regler befindet sich auf dem Fenster.

- Parameters:**
- pFenster - das Fenster, das die Komponente enthält
 - pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pAnfangswert - der Anfangswert des Reglers
 - pMinwert - der Minimalwert des Reglers
 - pMaxwert - der Maximalwert des Reglers

Regler

```
public Regler(Fenster pFenster,  
              int pStil,  
              int pAnfangswert,  
              int pMinwert,  
              int pMaxwert)
```

Der Regler wird erzeugt. Der Stil ist Ausrichtung.VERTICAL oder Ausrichtung.HORIZONTAL. Ausserdem werden

Anfangswert, Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe wird im Anwendungsprogramm festgelegt. Der Regler befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pStil - die Ausrichtung des Reglers (siehe Klasse Ausrichtung)
- pAnfangswert - der Anfangswert des Reglers
- pMinwert - der Minimalwert des Reglers
- pMaxwert - der Maximalwert des Reglers

Regler

```
public Regler(int pStil,  
              int pAnfangswert,  
              int pMinwert,  
              int pMaxwert)
```

Der Regler wird erzeugt. Der Stil ist Ausrichtung.VERTICAL oder Ausrichtung.HORIZONTAL. Ausserdem werden Anfangswert, Minimalwert und Maximalwert uebergeben. Position, Breite und Hoehe wird im Anwendungsprogramm festgelegt.

Parameters:

- pStil - die Ausrichtung des Reglers (siehe Klasse Ausrichtung)
- pAnfangswert - der Anfangswert des Reglers
- pMinwert - der Minimalwert des Reglers
- pMaxwert - der Maximalwert des Reglers

Method Detail

bekommtFokus

```
protected void bekommtFokus()
```

Der Regler erhaelt den Fokus.

geaendert

```
protected void geaendert()
```

Der Regler reagiert auf eine Bewegung.

maximum

```
public int maximum()
```

Das momentane Maximum des Reglers wird abgefragt.

Returns:

- das aktuelle Maximum des Reglers

minimum

```
public int minimum()
```

Das momentane Minimum des Reglers wird abgefragt.

Returns:

- das aktuelle Minimum des Reglers

setzeBearbeiterGeaendert

```
public void setzeBearbeiterGeaendert(String pBearbeiter)
```

Die Methode zur Bearbeitung des Geaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:

- pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Regler geaendert wurde.

setzeMaximum

public void **setzeMaximum**(int pWert)

Der Regler erhaelt eine neues Maximum.

Parameters:

pWert - das neue Maximum des Reglers

setzeMinimum

public void **setzeMinimum**(int pWert)

Der Regler erhaelt eine neues Minimum.

Parameters:

pWert - das neue Minimum des Reglers

setzeWert

public void **setzeWert**(int pWert)

Der Regler erhaelt eine neuen aktuellen Wert.

Parameters:

pWert - der neue Wert des Reglers

verliertFokus

protected void **verliertFokus**()

Der Regler verliert den Fokus.

wert

public int **wert**()

Die momentane Position (Wert) des Reglers wird abgefragt.

Returns:

der aktuelle Wert des Reglers

sum.komponenten

Class Schalter



All Implemented Interfaces:
[Serializable](#)

```
public class Schalter
extends Textkomponente
implements Serializable
```

Ein Schalter ist die Klasse fuer Java-Checkboxes. Schalter reagieren auf einen Klick mit der Maus und schalten sich ein bzw. aus.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary

Schalter (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)
Der Schalter wird erzeugt und bekommt den Text pAufschrift.

Schalter (double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift, String pGeklicktBearbeiter)
Der Schalter wird erzeugt und bekommt den Text pAufschrift.

Schalter (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pAufschrift)
Der Schalter wird erzeugt und bekommt den Text pAufschrift.

Method Summary

boolean	angeschaltet ()	Es wird zurueckgegeben, ob der Schalter angeschaltet ist.
---------	---------------------------------	---

protected void	fokusErhalten ()	Der Schalter erhaelt den Fokus.
----------------	----------------------------------	---------------------------------

protected void	fokusVerloren ()	Der Schalter verliert den Fokus.
----------------	----------------------------------	----------------------------------

String	inhaltAlsText ()	Die Beschriftung des Schalters wird als String zurueckgegeben.
------------------------	----------------------------------	--

void	schalteAn ()	Der Schalter wird angeschaltet.
------	------------------------------	---------------------------------

void	schalteAus ()	Der Schalter wird ausgeschaltet.
------	-------------------------------	----------------------------------

protected void	schalterGeklickt ()	Der Schalter reagiert auf einen Mausklick, indem er die beim Konstruktor als Parameter uebergebene Methode der Anwendung aufruft.
----------------	-------------------------------------	---

void	setzeBearbeiterGeklickt (String pBearbeiter) Die Methode zur Bearbeitung des Geklickt-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeInhalt (String pText) Der Schalter erhaelt eine neue Aufschrift.

Methods inherited from class sum.komponenten.Textkomponente
inhaltAlsGanzeZahl , inhaltAlsLangeGanzeZahl , inhaltAlsZahl , inhaltIstGanzeZahl , inhaltIstLangeGanzeZahl , inhaltIstText , inhaltIstZahl , init , schriftfarbe , schriftFarbe , setzeInhalt , setzeInhalt , setzeInhalt , setzeInhalt , setzeSchriftart , setzeSchriftArt , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftfarbe , setzeSchriftFarbe , setzeSchriftFarbe , setzeSchriftgroesse , setzeSchriftGroesse , setzeSchriftstil , setzeSchriftStil

Methods inherited from class sum.komponenten.Komponente
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Constructor Detail

Schalter

```
public Schalter(double pLinks,
                double pOben,
                double pBreite,
                double pHoehe,
                String pAufschrift)
```

Der Schalter wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der Name der Aktionsmethode, die beim Druucken des Schalteres aufgerufen wird, muss noch festgelegt werden. Der Schalter befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pAufschrift - der Text des Schalters

Schalter

```
public Schalter(double pLinks,
                double pOben,
                double pBreite,
                double pHoehe,
                String pAufschrift,
                String pGeklicktBearbeiter)
```

Der Schalter wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der letzte Parameter ist der Name der Aktionsmethode, die beim Druucken des Schalters aufgerufen wird.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pAufschrift - der Text des Schalters
 - pGeklicktBearbeiter - der Bezeichner des Dienstes der SuMAnwendung, der aufgerufen wird, wenn der Schalter gedrueckt wurde.

Schalter

```
public Schalter(Fenster pFenster,
                double pLinks,
                double pOben,
                double pBreite,
                double pHoehe,
```


[String](#) pAufschrift)

Der Schalter wird erzeugt und bekommt den Text pAufschrift. Position, Breite und Hoehe werden als Parameter uebergeben. Der Name der Aktionsmethode, die beim Druecken des Schalteres aufgerufen wird, muss noch festgelegt werden. Der Schalter befindet sich auf dem Fenster.

Parameters:

pFenster - das Fenster, das die Komponente enthält
pLinks - der Abstand der Komponente vom linken Fensterrand
pOben - der Abstand der Komponente vom oberen Fensterrand
pBreite - die Breite der Komponente
pHoehe - die Hoehe der Komponente
pAufschrift - der Text des Schalters

Method Detail

angeschaltet

```
public boolean angeschaltet()
```

Es wird zurueckgegeben, ob der Schalter angeschaltet ist.

Returns:

true, wenn der Schalter angeschaltet ist

fokusErhalten

```
protected void fokusErhalten()
```

Der Schalter erhaelt den Fokus.

fokusVerloren

```
protected void fokusVerloren()
```

Der Schalter verliert den Fokus.

inhaltAlsText

```
public String inhaltAlsText()
```

Die Beschriftung des Schalters wird als String zurueckgegeben.

Specified by:

[inhaltAlsText](#) in class [Textkomponente](#)

Returns:

der aktuelle Text des Schalters

schalteAn

```
public void schalteAn()
```

Der Schalter wird angeschaltet.

schalteAus

```
public void schalteAus()
```

Der Schalter wird ausgeschaltet.

schalterGeklickt

```
protected void schalterGeklickt()
```

Der Schalter reagiert auf einen Mausklick, indem er die beim Konstruktor als Parameter uebergebene Methode der Anwendung aufruft.

setzeBearbeiterGeklickt

public void **setzeBearbeiterGeklickt**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des Geklickt-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMAnwendung, der aufgerufen wird, wenn der Schalter gedrueckt wurde.

setzeInhalt

public void **setzeInhalt**([String](#) pText)

Der Schalter erhaelt eine neue Aufschrift.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der neue Text des Schalters

sum.komponenten

Class Tabelle



All Implemented Interfaces:
[Serializable](#), [ScrollPaneConstants](#)

```
public class Tabelle
extends Komponente
implements Serializable, ScrollPaneConstants
```

Eine Tabelle ist die eingedeutschte Klasse fuer Java-JTables. Tabellen haben Zeilen, Spalten und Zellen. Spalten besitzen Überschriften, die auch ausgeblendet werden können.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
protected String	zAktuellFont
protected Font	zSchriftArt
protected int	zSchriftGroesse
protected int	zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Fields inherited from interface javax.swing. ScrollPaneConstants
COLUMN_HEADER , HORIZONTAL_SCROLLBAR , HORIZONTAL_SCROLLBAR_ALWAYS , HORIZONTAL_SCROLLBAR_AS_NEEDED , HORIZONTAL_SCROLLBAR_NEVER , HORIZONTAL_SCROLLBAR_POLICY , LOWER_LEADING_CORNER , LOWER_LEFT_CORNER , LOWER_RIGHT_CORNER , LOWER_TRAILING_CORNER , ROW_HEADER , UPPER_LEADING_CORNER , UPPER_LEFT_CORNER , UPPER_RIGHT_CORNER , UPPER_TRAILING_CORNER , VERTICAL_SCROLLBAR , VERTICAL_SCROLLBAR_ALWAYS , VERTICAL_SCROLLBAR_AS_NEEDED , VERTICAL_SCROLLBAR_NEVER , VERTICAL_SCROLLBAR_POLICY , VIEWPORT

Constructor Summary
Tabelle (double pLinks, double pOben, double pBreite, double pHoehe, int pZeilen, int pSpalten) Die Tabelle wird erzeugt und enthaelt den uebergebenen Text.
Tabelle (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, int pZeilen, int pSpalten) Die Tabelle wird erzeugt und enthaelt den uebergebenen Text.

Method Summary
protected void bekommtFokus () Der Zeilenbereich erhaelt den Fokus.
void entferneSpalteAn (int pSpalte) Die Spalte an Position pSpalte wird entfernt.
void entferneZeileAn (int pZeile) Die Zeile an Position pZeile wird entfernt.
void fuegeNeueSpalteEinAn (int pSpalte)

	Eine neue Spalte wird Position pSpalte eingefuegt.
void	fuegeNeueZeileEinAn (int pZeile) Eine neue Zeile wird Position pZeile eingefuegt.
void	haengeAnAn (String pText, int pZeile, int pSpalte) Der Text pText wird in Zelle pSpalte, pZeile angehaengt.
void	haengeNeueSpalteAn () Eine neue Spalte wird angehaengt.
void	haengeNeueZeileAn () Eine neue Zeile wird angehaengt.
int	inhaltAlsGanzeZahlAn (int pZeile, int pSpalte) Der Inhalt einer Zelle der Tabelle wird als ganze Zahl zurueckgegeben.
String	inhaltAlsTextAn (int pZeile, int pSpalte) Der Inhalt einer Zelle der Tabelle wird als String zurueckgegeben.
double	inhaltAlsZahlAn (int pZeile, int pSpalte) Der Inhalt einer Zelle der Tabelle wird als Kommazahl zurueckgegeben.
protected void	inhaltGeaendert () Die Tabelle reagiert auf Veraenderungen.
boolean	inhaltIstGanzeZahlAn (int pZeile, int pSpalte) Es wird zurueckgegeben, ob die Textkomponente eine ganze Zahl ist.
boolean	inhaltIstTextAn (int pZeile, int pSpalte) Es wird zurueckgegeben, ob die Zelle an Position pZeile, pSpalte keine (ganze) Zahl ist.
boolean	inhaltIstZahlAn (int pZeile, int pSpalte) Es wird zurueckgegeben, ob die Textkomponente eine Kommazahl ist.
boolean	istSpalteMarkiert (int pSpalte) Es wird zurueckgegeben, ob die Spalte markiert ist.
boolean	istZeileMarkiert (int pZeile) Es wird zurueckgegeben, ob die Zeile markiert ist.
boolean	istZelleMarkiert (int pZeile, int pSpalte) Es wird zurueckgegeben, ob die Zelle markiert ist.
void	markiereAlles () Alle Zellen werden markiert.
void	markiereNichts () Alle Zeilen werden nicht markiert.
protected void	markierungGeaendert () Die Tabelle reagiert auf Veraenderungen der Markierung.
void	setzeBearbeiterInhaltGeaendert (String pBearbeiter) Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeBearbeiterMarkierungGeaendert (String pBearbeiter) Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeGroesse (double pBreite, double pHoehe) Die Tabelle erhaelt eine neue Groesse.
void	setzeInhaltAn (char pZeichen, int pZeile, int pSpalte) Das Zeichen pZeichen wird in Zelle pSpalte, pZeile eingefuegt.
void	setzeInhaltAn (double pZahl, int pZeile, int pSpalte) Die Zahl pZahl wird in Zelle pSpalte, pZeile eingefuegt.
void	setzeInhaltAn (int pZahl, int pZeile, int pSpalte) Die ganze Zahl pZahl wird in Zelle pSpalte, pZeile eingefuegt.
void	setzeInhaltAn (String pText, int pZeile, int pSpalte) Der Text pText wird in Zelle pSpalte, pZeile eingefuegt.
void	setzeMarkierteSpalten (int pAnfang, int pEnde) Die Spalten von pAnfang bis pEnde werden zusaetzlich markiert.
void	setzeMarkierteZeilen (int pAnfang, int pEnde) Die Zeilen von pAnfang bis pEnde werden zusaetzlich markiert.
void	setzeMarkierteZelle (int pZeile, int pSpalte) Die Zelle pZeile, pSpalte wird zusaetzlich markiert.
void	setzePosition (double pWohinH, double pWohinV) Die Tabelle erhaelt eine neue Position (Ecke oben links).
void	setzeSchriftart (String pSchriftart) Die Schriftart der Tabelle wird veraendert

void	setzeSchriftArt (String pSchriftart) Die Schriftart der Tabelle wird veraendert
void	setzeSchriftfarbe (Color pFarbe) Die Schriftfarbe der Tabelle wird veraendert
void	setzeSchriftFarbe (Color pFarbe) Die Schriftfarbe der Tabelle wird veraendert
void	setzeSchriftfarbe (int pFarbe) Die Schriftfarbe der Tabelle wird veraendert
void	setzeSchriftFarbe (int pFarbe) Die Schriftfarbe der Tabelle wird veraendert
void	setzeSchriftgroesse (int pGroesse) Die Schriftgroesse der Tabelle wird veraendert
void	setzeSchriftGroesse (int pGroesse) Die Schriftgroesse der Tabelle wird veraendert
void	setzeSchriftstil (int pStil) Der Schriftstil der Tabelle wird veraendert
void	setzeSchriftStil (int pStil) Der Schriftstil der Tabelle wird veraendert
void	setzeSpaltenanzahl (int pAnzahl) Die Anzahl der Spalten in der Tabelle wird veraendert.
void	setzeSpaltenbreite (int pBreite) Die Spalten der Tabelle erhalten eine neue Breite.
void	setzeSpaltentitelAn (String pText, int pSpalte) Der Text pText wird alsTitel fuer Spalte pSpalte eingefuegt.
void	setzeZeilenanzahl (int pAnzahl) Die Anzahl der Zeilen in der Tabelle wird veraendert.
void	setzeZeilenhoehe (int pHoehe) Die Zeilen der Tabelle erhalten eine neue Hoehe.
int	spaltenanzahl () Die Anzahl der Spalten in der Tabelle wird zurueckgegeben.
String	spaltentitel (int pSpalte) Der Spaltentitel der Spalte pSpalte wird zurueckgegeben.
protected void	verliertFokus () Die Tabelle verliert den Fokus.
int	zeilenanzahl () Die Anzahl der Zeilen im der Tabelle wird zurueckgegeben.

Methods inherited from class <code>sum.komponenten.Komponente</code>
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeHinweis , verstecke , zeige

Methods inherited from class <code>java.lang.Object</code>
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

zAktuellFont

protected [String](#) **zAktuellFont**

zSchriftArt

protected [Font](#) **zSchriftArt**

zSchriftGroesse

protected int **zSchriftGroesse**

zSchriftStil

Constructor Detail

Tabelle

```
public Tabelle(double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               int pZeilen,
               int pSpalten)
```

Die Tabelle wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe sowie die Anzahl der Zeilen und Spalten werden als Parameter uebergeben. Die Tabelle befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pZeilen - die Anzahl der Zeilen der Tabelle
 - pSpalten - die Anzahl der Spalten der Tabelle

Tabelle

```
public Tabelle(Fenster pFenster,
               double pLinks,
               double pOben,
               double pBreite,
               double pHoehe,
               int pZeilen,
               int pSpalten)
```

Die Tabelle wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe sowie die Anzahl der Zeilen und Spalten werden als Parameter uebergeben. Die Tabelle befindet sich auf dem Fenster.

- Parameters:**
- pFenster - das Fenster, das die Komponente enthält
 - pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pZeilen - die Anzahl der Zeilen der Tabelle
 - pSpalten - die Anzahl der Spalten der Tabelle

Method Detail

bekommtFokus

```
protected void bekommtFokus()
```

Der Zeilenbereich erhaelt den Fokus.

entferneSpalteAn

```
public void entferneSpalteAn(int pSpalte)
```

Die Spalte an Position pSpalte wird entfernt.

- Parameters:**
- pSpalte - die Position, wo die Spalte entfernt wird

entferneZeileAn

```
public void entferneZeileAn(int pZeile)
```

Die Zeile an Position pZeile wird entfernt.

- Parameters:**
- pZeile - die Position, wo die Zeile entfernt wird

fuegeNeueSpalteEinAn

public void **fuegeNeueSpalteEinAn**(int pSpalte)

Eine neue Spalte wird Position pSpalte eingefuegt.

Parameters:

pSpalte - die Position, wo eine neue Spalte eingefuegt wird

fuegeNeueZeileEinAn

public void **fuegeNeueZeileEinAn**(int pZeile)

Eine neue Zeile wird Position pZeile eingefuegt.

Parameters:

pZeile - die Position, wo eine neue Zeile eingefuegt wird

haengeAnAn

public void **haengeAnAn**([String](#) pText,
int pZeile,
int pSpalte)

Der Text pText wird in Zelle pSpalte, pZeile angehaengt.

Parameters:

pText - der angehaengte Text der Zelle
pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

haengeNeueSpalteAn

public void **haengeNeueSpalteAn**()

Eine neue Spalte wird angehaengt.

haengeNeueZeileAn

public void **haengeNeueZeileAn**()

Eine neue Zeile wird angehaengt.

inhaltAlsGanzeZahlAn

public int **inhaltAlsGanzeZahlAn**(int pZeile,
int pSpalte)
throws [ArithmeticException](#)

Der Inhalt einer Zelle der Tabelle wird als ganze Zahl zurueckgegeben.

Parameters:

pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

Returns:

der Inhalt der Zelle als ganze Zahl

Throws:

[ArithmeticException](#)

inhaltAlsTextAn

public [String](#) **inhaltAlsTextAn**(int pZeile,
int pSpalte)

Der Inhalt einer Zelle der Tabelle wird als String zurueckgegeben.

Parameters:

pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

Returns:

der Inhalt der Zelle als Zeichenkette (String)

inhaltAlsZahlAn

```
public double inhaltAlsZahlAn(int pZeile,  
                             int pSpalte)  
    throws ArithmeticException
```

Der Inhalt einer Zelle der Tabelle wird als Kommazahl zurueckgegeben.

Parameters:

pZeile - die Zeilennummer der Zelle

pSpalte - die Spaltennummer der Zelle

Returns:

der Inhalt der Zelle als Kommazahl

Throws:

[ArithmeticException](#)

inhaltGeaendert

```
protected void inhaltGeaendert()
```

Die Tabelle reagiert auf Veraenderungen.

inhaltIstGanzeZahlAn

```
public boolean inhaltIstGanzeZahlAn(int pZeile,  
                                     int pSpalte)
```

Es wird zurueckgegeben, ob die Textkomponente eine ganze Zahl ist.

Parameters:

pZeile - die Zeilennummer der Zelle

pSpalte - die Spaltennummer der Zelle

Returns:

true, wenn der Inhalt der Zelle eine ganze Zahl ist

inhaltIstTextAn

```
public boolean inhaltIstTextAn(int pZeile,  
                                int pSpalte)
```

Es wird zurueckgegeben, ob die Zelle an Position pZeile, pSpalte keine (ganze) Zahl ist.

Parameters:

pZeile - die Zeilennummer der Zelle

pSpalte - die Spaltennummer der Zelle

Returns:

true, wenn der Inhalt der Zelle keine Zahl ist

inhaltIstZahlAn

```
public boolean inhaltIstZahlAn(int pZeile,  
                                int pSpalte)
```

Es wird zurueckgegeben, ob die Textkomponente eine Kommazahl ist.

Parameters:

pZeile - die Zeilennummer der Zelle

pSpalte - die Spaltennummer der Zelle

Returns:

true, wenn der Inhalt der Zelle eine Zahl ist

istSpalteMarkiert

```
public boolean istSpalteMarkiert(int pSpalte)
```

Es wird zurueckgegeben, ob die Spalte markiert ist.

Parameters:

pSpalte - die Nummer der Spalte
Returns:
true, wenn die Spalte markiert ist

istZeileMarkiert

```
public boolean istZeileMarkiert(int pZeile)
```

Es wird zurueckgegeben, ob die Zeile markiert ist.

Parameters:

pZeile - die Nummer der Zeile

Returns:

true, wenn die Zeile markiert ist

istZelleMarkiert

```
public boolean istZelleMarkiert(int pZeile,  
                                int pSpalte)
```

Es wird zurueckgegeben, ob die Zelle markiert ist.

Parameters:

pZeile - die Nummer der Zeile

pSpalte - die Nummer der Spalte

Returns:

true, wenn die Zelle markiert ist

markiereAlles

```
public void markiereAlles()
```

Alle Zellen werden markiert.

markiereNichts

```
public void markiereNichts()
```

Alle Zeilen werden nicht markiert.

markierungGeaendert

```
protected void markierungGeaendert()
```

Die Tabelle reagiert auf Veraenderungen der Markierung.

setzeBearbeiterInhaltGeaendert

```
public void setzeBearbeiterInhaltGeaendert(String pBearbeiter)
```

Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn sich der Inhalt der Tabelle aendert.

setzeBearbeiterMarkierungGeaendert

```
public void setzeBearbeiterMarkierungGeaendert(String pBearbeiter)
```

Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn sich die Markierung der Tabelle aendert.

setzeGroesse

```
public void setzeGroesse(double pBreite,  
                        double pHoehe)
```

Die Tabelle erhaelt eine neue Groesse.

Overrides:
[setzeGroesse](#) in class [Komponente](#)

Parameters:
pBreite - die neue Breite der Komponente
pHoehe - die neue Hoehe der Komponente

setzeInhaltAn

```
public void setzeInhaltAn(char pZeichen,  
                        int pZeile,  
                        int pSpalte)
```

Das Zeichen pZeichen wird in Zelle pSpalte, pZeile eingefuegt.

Parameters:
pZeichen - der neue Inhalt der Zelle
pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

setzeInhaltAn

```
public void setzeInhaltAn(double pZahl,  
                        int pZeile,  
                        int pSpalte)
```

Die Zahl pZahl wird in Zelle pSpalte, pZeile eingefuegt.

Parameters:
pZahl - der neue Inhalt der Zelle
pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

setzeInhaltAn

```
public void setzeInhaltAn(int pZahl,  
                        int pZeile,  
                        int pSpalte)
```

Die ganze Zahl pZahl wird in Zelle pSpalte, pZeile eingefuegt.

Parameters:
pZahl - der neue Inhalt der Zelle
pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

setzeInhaltAn

```
public void setzeInhaltAn(String pText,  
                        int pZeile,  
                        int pSpalte)
```

Der Text pText wird in Zelle pSpalte, pZeile eingefuegt.

Parameters:
pText - der neue Inhalt der Zelle
pZeile - die Zeilennummer der Zelle
pSpalte - die Spaltennummer der Zelle

setzeMarkierteSpalten

```
public void setzeMarkierteSpalten(int pAnfang,  
                                int pEnde)
```

Die Spalten von pAnfang bis pEnde werden zusaetzlich markiert.

Parameters:

pAnfang - die Nummer der ersten Spalte
pEnde - die Nummer der letzten Spalte

setzeMarkierteZeilen

```
public void setzeMarkierteZeilen(int pAnfang,  
                                int pEnde)
```

Die Zeilen von pAnfang bis pEnde werden zusaetzlich markiert.

Parameters:

pAnfang - die Nummer der ersten Zeile
pEnde - die Nummer der letzten Zeile

setzeMarkierteZelle

```
public void setzeMarkierteZelle(int pZeile,  
                                int pSpalte)
```

Die Zelle pZeile, pSpalte wird zusaetzlich markiert.

Parameters:

pZeile - die Nummer der Zeile
pSpalte - die Nummer der Spalte

setzePosition

```
public void setzePosition(double pWohinH,  
                           double pWohinV)
```

Die Tabelle erhaelt eine neue Position (Ecke oben links).

Overrides:

[setzePosition](#) in class [Komponente](#)

Parameters:

pWohinH - die neue horizontale Position der linken oberen Ecke
pWohinV - die neue vertikale Position der linken oberen Ecke

setzeSchriftart

```
public void setzeSchriftart(String pSchriftart)
```

Die Schriftart der Tabelle wird veraendert

Parameters:

pSchriftart - die neue Schriftart der Tabelle (siehe Klasse Schrift)

setzeSchriftArt

```
public void setzeSchriftArt(String pSchriftart)
```

Die Schriftart der Tabelle wird veraendert

Parameters:

pSchriftart - die neue Schriftart der Tabelle (siehe Klasse Schrift)

setzeSchriftfarbe

```
public void setzeSchriftfarbe(Color pFarbe)
```

Die Schriftfarbe der Tabelle wird veraendert

Parameters:

pFarbe - die neue Schriftfarbe der Tabelle

setzeSchriftFarbe

```
public void setzeSchriftFarbe(Color pFarbe)
```

Die Schriftfarbe der Tabelle wird veraendert

Parameters:
pFarbe - die neue Schriftfarbe der Tabelle

setzeSchriftfarbe

public void **setzeSchriftfarbe**(int pFarbe)

Die Schriftfarbe der Tabelle wird veraendert

Parameters:
pFarbe - die neue Schriftfarbe der Tabelle

setzeSchriftFarbe

public void **setzeSchriftFarbe**(int pFarbe)

Die Schriftfarbe der Tabelle wird veraendert

Parameters:
pFarbe - die neue Schriftfarbe der Tabelle

setzeSchriftgroesse

public void **setzeSchriftgroesse**(int pGroesse)

Die Schriftgroesse der Tabelle wird veraendert

Parameters:
pGroesse - die neue Schriftgroesse der Tabelle

setzeSchriftGroesse

public void **setzeSchriftGroesse**(int pGroesse)

Die Schriftgroesse der Tabelle wird veraendert

Parameters:
pGroesse - die neue Schriftgroesse der Tabelle

setzeSchriftstil

public void **setzeSchriftstil**(int pStil)

Der Schriftstil der Tabelle wird veraendert

Parameters:
pStil - der neue Schriftstil der Tabelle (siehe Klasse Schrift)

setzeSchriftStil

public void **setzeSchriftStil**(int pStil)

Der Schriftstil der Tabelle wird veraendert

Parameters:
pStil - der neue Schriftstil der Tabelle (siehe Klasse Schrift)

setzeSpaltenanzahl

public void **setzeSpaltenanzahl**(int pAnzahl)

Die Anzahl der Spalten in der Tabelle wird veraendert.

Parameters:
pAnzahl - die neue Anzahl der Spalten der Tabelle

setzeSpaltenbreite

```
public void setzeSpaltenbreite(int pBreite)
```

Die Spalten der Tabelle erhalten eine neue Breite.

Parameters:
pBreite - die neue Spaltenbreite

setzeSpaltentitelAn

```
public void setzeSpaltentitelAn(String pText,  
                                int pSpalte)
```

Der Text pText wird alsTitel fuer Spalte pSpalte eingefuegt.

Parameters:
pText - der Titel der Spalte
pSpalte - die Position der Spalte fuer den Titel

setzeZeilenanzahl

```
public void setzeZeilenanzahl(int pAnzahl)
```

Die Anzahl der Zeilen in der Tabelle wird veraendert.

Parameters:
pAnzahl - die neue Anzahl der Zeilen der Tabelle

setzeZeilenhoehe

```
public void setzeZeilenhoehe(int pHoehe)
```

Die Zeilen der Tabelle erhalten eine neue Hoehe.

Parameters:
pHoehe - die neue Zeilenhoehe

spaltenanzahl

```
public int spaltenanzahl()
```

Die Anzahl der Spalten in der Tabelle wird zurueckgegeben.

Returns:
die Anzahl der Spalten der Tabelle

spaltentitel

```
public String spaltentitel(int pSpalte)
```

Der Spaltentitel der Spalte pSpalte wird zurueckgegeben.

Parameters:
pSpalte - die Position der Spalte fuer den Titel

Returns:
der Spaltentitel

verliertFokus

```
protected void verliertFokus()
```

Die Tabelle verliert den Fokus.

zeilenanzahl

```
public int zeilenanzahl()
```

Die Anzahl der Zeilen im der Tabelle wird zurueckgegeben.

Returns:
die Anzahl der Zeilen der Tabelle

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.komponenten

Class Textbereich

[java.lang.Object](#)



All Implemented Interfaces:

[Serializable](#), [ScrollPaneConstants](#)

Direct Known Subclasses:

[Zeichenbereich](#), [Zeilenbereich](#)

```
public abstract class Textbereich
extends Markierungskomponente
implements Serializable, ScrollPaneConstants
```

Ein Textbereich ist die Klasse fuer Java-TextAreas. Textbereiche sind Editoren mit einer festen Breite und Hoehe. Sie reagieren auf Texteingaben und Mausklicks. Wenn etwas in einen Textbereich getippt werden soll, muss es vorher mit einem Mausklick aktiviert werden, d.h. der Textbereich muss den Fokus haben. Es gib zwei Unterklassen von Textbereich: Zeichenbereich und Zeilenbereich. Der Textbereich ist zeichenorientiert, der Zeilenbereich ist zeilenorientiert. Der Textbereich ist abstrakt wird normalerweise nicht benutzt.

Version:

7.3 vom 11.10.2011

Author:

Bernard Schriek

See Also:

[Serialized Form](#)

Field Summary

protected JScrollPane	hatScrollPane
--	-------------------------------

Fields inherited from class sum.komponenten.[Markierungskomponente](#)

[zInhaltGeaendertBearbeiter](#), [zMarkierungGeaendertBearbeiter](#)

Fields inherited from class sum.komponenten.[Textkomponente](#)

[zAktuellFont](#), [zSchriftArt](#), [zSchriftGroesse](#), [zSchriftStil](#)

Fields inherited from class sum.komponenten.[Komponente](#)

[hatComponent](#), [kenntFenster](#)

Fields inherited from interface javax.swing.[ScrollPaneConstants](#)

[COLUMN_HEADER](#), [HORIZONTAL_SCROLLBAR](#), [HORIZONTAL_SCROLLBAR_ALWAYS](#), [HORIZONTAL_SCROLLBAR_AS_NEEDED](#), [HORIZONTAL_SCROLLBAR_NEVER](#), [HORIZONTAL_SCROLLBAR_POLICY](#), [LOWER_LEADING_CORNER](#), [LOWER_LEFT_CORNER](#), [LOWER_RIGHT_CORNER](#), [LOWER_TRAILING_CORNER](#), [ROW_HEADER](#), [UPPER_LEADING_CORNER](#), [UPPER_LEFT_CORNER](#), [UPPER_RIGHT_CORNER](#), [UPPER_TRAILING_CORNER](#), [VERTICAL_SCROLLBAR](#), [VERTICAL_SCROLLBAR_ALWAYS](#), [VERTICAL_SCROLLBAR_AS_NEEDED](#), [VERTICAL_SCROLLBAR_NEVER](#), [VERTICAL_SCROLLBAR_POLICY](#), [VIEWPORT](#)

Constructor Summary

[Textbereich](#)(double pLinks, double pOben, double pBreite, double pHoehe)
Der Textbereich wird erzeugt und enthaelt den uebergebenen Text.

[Textbereich](#)([Fenster](#) pFenster, double pLinks, double pOben, double pBreite, double pHoehe)
Der Textbereich wird erzeugt und enthaelt den uebergebenen Text.

Method Summary

abstract void	fuegeEin (String pText, int pStelle) Der Text pText wird an Position pStelle eingefuegt.
abstract void	haengeAn (char pZeichen) Das Zeichen pZeichen wird am Ende angehaengt.
abstract void	haengeAn (double pZahl) Das Kommazahl pZahl wird am Ende angehaengt.
abstract void	haengeAn (int pZahl) Das ganze Zahl pZahl wird am Ende angehaengt.
abstract void	haengeAn (String pText) Der Text pText wird am Ende angehaengt.
protected void	LerneKomponenteKennen (Bildschirm pFenster, JComponent pKomponente) Die Markierungskomponente kennt die eigentliche TextComponent aus der Java-Bibliothek.
void	setzeGroesse (double pBreite, double pHoehe) Die Komponente erhaelt eine neue Groesse.
void	setzePosition (double pWohinH, double pWohinV) Die Komponente erhaelt eine neue Position (Ecke oben links).

Methods inherited from class `sum.komponenten.Markierungskomponente`

[istMarkiert](#), [loesche](#), [loescheAlles](#), [loescheMarkierung](#), [markiereAlles](#), [markiereNichts](#), [markierterInhalt](#), [markierungGeaendert](#), [markierungsAnfang](#), [markierungsEnde](#), [setzeBearbeiterInhaltGeaendert](#), [setzeBearbeiterMarkierungGeaendert](#), [setzeMarkierung](#), [teillinhalt](#), [teilInhalt](#)

Methods inherited from class `sum.komponenten.Textkomponente`

[inhaltAlsGanzeZahl](#), [inhaltAlsLangeGanzeZahl](#), [inhaltAlsText](#), [inhaltAlsZahl](#), [inhaltIstGanzeZahl](#), [inhaltIstLangeGanzeZahl](#), [inhaltIstText](#), [inhaltIstZahl](#), [init](#), [schriftfarbe](#), [schriftFarbe](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeSchriftart](#), [setzeSchriftArt](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftFarbe](#), [setzeSchriftgroesse](#), [setzeSchriftGroesse](#), [setzeSchriftstil](#), [setzeSchriftStil](#)

Methods inherited from class `sum.komponenten.Komponente`

[aktiviere](#), [besitztFokus](#), [breite](#), [deaktiviere](#), [farbe](#), [fokusErhaltenBearbeiter](#), [fokusVerlorenBearbeiter](#), [gibFrei](#), [hoehe](#), [init](#), [istAktiv](#), [istSichtbar](#), [links](#), [oben](#), [setzeBearbeiterFokusErhalten](#), [setzeBearbeiterFokusVerloren](#), [setzeFarbe](#), [setzeFarbe](#), [setzeFokus](#), [setzeFokusWert](#), [setzeHinweis](#), [verstecke](#), [zeige](#)

Methods inherited from class `java.lang.Object`

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

hatScrollPane

protected [JScrollPane](#) `hatScrollPane`

Constructor Detail

Textbereich

```
public Textbereich(double pLinks,
                   double pOben,
                   double pBreite,
                   double pHoehe)
```

Der Textbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Textbereich befindet sich auf dem Bildschirm.

- Parameters:
- pLinks

- der Abstand der Komponente vom linken Fensterrand
- pOben

- der Abstand der Komponente vom oberen Fensterrand
- pBreite

- die Breite der Komponente
- pHoehe

- die Hoehe der Komponente

Textbereich

```
public Textbereich(Fenster pFenster,
                   double pLinks,
                   double pOben,
```



```
double pBreite,  
double pHoehe)
```

Der Textbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Textbereich befindet sich auf dem Fenster.

Parameters:

pFenster - das Fenster, das die Komponente enthaelt
pLinks - der Abstand der Komponente vom linken Fensterrand
pOben - der Abstand der Komponente vom oberen Fensterrand
pBreite - die Breite der Komponente
pHoehe - die Hoehe der Komponente

Method Detail

fuegeEin

```
public abstract void fuegeEin(String pText,  
                               int pStelle)
```

Der Text pText wird an Position pStelle eingefuegt.

Specified by:

[fuegeEin](#) in class [Markierungskomponente](#)

Parameters:

pText - der Text, der eingefuegt wird
pStelle - die Stelle, wo der Text eingefuegt wird

haengeAn

```
public abstract void haengeAn(char pZeichen)
```

Das Zeichen pZeichen wird am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Markierungskomponente](#)

Parameters:

pZeichen - das Zeichen, das ans Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(double pZahl)
```

Das Kommazahl pZahl wird am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Markierungskomponente](#)

Parameters:

pZahl - die Zahl, die ans Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(int pZahl)
```

Das ganze Zahl pZahl wird am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Markierungskomponente](#)

Parameters:

pZahl - die Zahl, die ans Ende angehaengt wird

haengeAn

```
public abstract void haengeAn(String pText)
```

Der Text pText wird am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Markierungskomponente](#)

Parameters:

pText - der Text, der ans Ende angehaengt wird

lerneKomponenteKennen

```
protected void lerneKomponenteKennen(Bildschirm pFenster,  
                                       JComponent pKomponente)
```

Die Markierungskomponente kennt die eigentliche TextComponent aus der Java-Bibliothek.

Overrides:

[lerneKomponenteKennen](#) in class [Komponente](#)

setzeGroesse

```
public void setzeGroesse(double pBreite,  
                        double pHoehe)
```

Die Komponente erhaelt eine neue Groesse.

Overrides:

[setzeGroesse](#) in class [Komponente](#)

Parameters:

pBreite - die neue Breite der Komponente
pHoehe - die neue Hoehe der Komponente

setzePosition

```
public void setzePosition(double pWohinH,  
                        double pWohinV)
```

Die Komponente erhaelt eine neue Position (Ecke oben links).

Overrides:

[setzePosition](#) in class [Komponente](#)

Parameters:

pWohinH - die neue horizontale Position der linken oberen Ecke
pWohinV - die neue vertikale Position der linken oberen Ecke

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.komponenten

Class Textfeld

[java.lang.Object](#)



All Implemented Interfaces:

[Serializable](#)

Direct Known Subclasses:

[Kennwortfeld](#)

```
public class Textfeld
extends Markierungskomponente
implements Serializable
```

Ein Textfeld ist die Klasse fuer Java-TextFields. Textfelder sind Editoren mit einer festen Breite und Hoehe. Im Gegensatz zu Textbereichen haben sie keinen Rollbalken und bestehen nur aus einer einzigen Zeile. Sie reagieren auf Texteingaben und Mausklicks. Wenn etwas in ein Textfeld getippt werden soll, muss es vorher mit einem Mausklick aktiviert werden, d.h. das Textfeld muss den Fokus haben.

Version:

7.3 vom 11.10.2011

Author:

Bernard Schriek, Horst Hildebrecht

See Also:

[Serialized Form](#)

Field Summary

protected JTextField	hatTextField
protected String	zEingabeBestaetigtBearbeiter

Fields inherited from class sum.komponenten.Markierungskomponente
zInhaltGeaendertBearbeiter , zMarkierungGeaendertBearbeiter

Fields inherited from class sum.komponenten.Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten.Komponente
hatComponent , kenntFenster

Constructor Summary

Textfeld ()	Dummy-Konstruktor fuer die Unterklasse Kennwortfeld.
Textfeld (double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Das Textfeld wird erzeugt und enthaelt den uebergebenen Text.
Textfeld (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Das Textfeld wird erzeugt und enthaelt den uebergebenen Text.

Method Summary

protected void	bekommtFokus ()	Das Textfeld erhaelt den Fokus.
protected		

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

hatTextField

protected JTextField hatTextField

zEingabeBestaetigtBearbeiter

protected String zEingabeBestaetigtBearbeiter

Constructor Detail

Textfeld

public Textfeld()

Dummy-Konstruktor für die Unterklasse Kennwortfeld.

Textfeld

public Textfeld(double pLinks,
double pOben,
double pBreite,
double pHoehe,
String pText)

Das Textfeld wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Das Textfeld befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Textfeld

public Textfeld(Fenster pFenster,
double pLinks,
double pOben,
double pBreite,
double pHoehe,
String pText)

Das Textfeld wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Das Textfeld befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthält
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Method Detail

bekommtFokus

protected void bekommtFokus()

Das Textfeld erhaelt den Fokus.

eingabeBestaetigt

protected void **eingabeBestaetigt**()

Das Textfeld reagiert auf die Eingabebestaetigung (Enter).

fuegeEin

public void **fuegeEin**([String](#) pText,
int pStelle)

Der Text pText wird an Position pStelle eingefuegt.

Specified by:
[fuegeEin](#) in class [Markierungskomponente](#)

Parameters:
pText - der Text, der eingefuegt wird
pStelle - die Stelle, wo der Text eingefuegt wird

haengeAn

public void **haengeAn**(char pZeichen)

Das Zeichen pZeichen wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Markierungskomponente](#)

Parameters:
pZeichen - das Zeichen, das am Ende angehaengt wird

haengeAn

public void **haengeAn**(double pZahl)

Die Kommazahl pZahl wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Markierungskomponente](#)

Parameters:
pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

public void **haengeAn**(int pZahl)

Das ganze Zahl pZahl wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Markierungskomponente](#)

Parameters:
pZahl - die Zahl, die am Ende angehaengt wird

haengeAn

public void **haengeAn**([String](#) pText)

Der Text pText wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Markierungskomponente](#)

Parameters:
pText - der Text, der am Ende angehaengt wird

inhaltAlsText

```
public String inhaltAlsText()
```

Der Text des Textfeldes wird als String zurueckgegeben.

Specified by:
[inhaltAlsText](#) in class [Textkomponente](#)

Returns:
der Inhalt der Komponente als Zeichenkette (String)

inhaltGeaendert

```
protected void inhaltGeaendert()
```

Das Textfeld reagiert auf Veraenderungen.

istMarkiert

```
public boolean istMarkiert()
```

Es wird zurueckgegeben, ob der Text markiert ist.

Specified by:
[istMarkiert](#) in class [Markierungskomponente](#)

Returns:
true, wenn Text markiert ist

loesche

```
public void loesche(int pAnfang,  
                   int pEnde)
```

Der durch die Parameter bestimmte Text wird geloescht.

Specified by:
[loesche](#) in class [Markierungskomponente](#)

Parameters:
pAnfang - erste Stelle
pEnde - letzte Stelle

loescheAlles

```
public void loescheAlles()
```

Der gesamte Text wird geloescht.

Overrides:
[loescheAlles](#) in class [Markierungskomponente](#)

loescheMarkierung

```
public void loescheMarkierung()
```

Der markierte Text wird geloescht.

Specified by:
[loescheMarkierung](#) in class [Markierungskomponente](#)

markiereAlles

```
public void markiereAlles()
```

Der gesamte Text wird markiert.

Specified by:
[markiereAlles](#) in class [Markierungskomponente](#)

markiereNichts

```
public void markiereNichts()
```

Der gesamte Text wird nicht markiert.

Specified by:
[markiereNichts](#) in class [Markierungskomponente](#)

markierterInhalt

```
public String markierterInhalt()
```

Der im Textfeld markierte Text wird geliefert.

Specified by:
[markierterInhalt](#) in class [Markierungskomponente](#)

Returns:
der markierte Text

markierungGeaendert

```
protected void markierungGeaendert()
```

Das Textfeld reagiert auf Veraenderungen der Markierung.

Specified by:
[markierungGeaendert](#) in class [Markierungskomponente](#)

markierungsAnfang

```
public int markierungsAnfang()
```

Es wird zurueckgegeben, von wo an der Text markiert ist.

Specified by:
[markierungsAnfang](#) in class [Markierungskomponente](#)

Returns:
die erste markierte Stelle

markierungsEnde

```
public int markierungsEnde()
```

Es wird zurueckgegeben, bis wohin der Text markiert ist.

Specified by:
[markierungsEnde](#) in class [Markierungskomponente](#)

Returns:
die letzte markierte Stelle

setzeAusrichtung

```
public void setzeAusrichtung(int pAusrichtung)
```

Die Ausrichtung des Texts im Textfeld wird geaendert.

Parameters:
pAusrichtung - die neue Ausrichtung des Textes (siehe Klasse Ausrichtung)

setzeBearbeiterEingabeBestaetigt

```
public void setzeBearbeiterEingabeBestaetigt(String pBearbeiter)
```

Die Methode zur Bearbeitung des EingabeBestaetigt-Ereignisses in der Ereignisanwendung wird festgelegt.

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuManwendung, der aufgerufen wird, wenn in der Komponente Enter gedrueckt wurde.

setzeInhalt

public void **setzeInhalt**([String](#) pText)

Ds Textfeld erhaelt einen neuen Inhalt.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der Text, der als neuer Inhalt gesetzt wird

setzeMarkierung

public void **setzeMarkierung**(int pAnfang,
int pEnde)

Der Text zwischen den Zeichen an Position pAnfang und pEnde wird markiert.

Specified by:

[setzeMarkierung](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Stelle
pEnde - letzte Stelle

teihinhalt

public [String](#) **teihinhalt**(int pAnfang,
int pEnde)

Der Inhalt des Textfeldes von pAnfang bis pEnde wird als String zurueckgegeben.

Specified by:

[teihinhalt](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Stelle
pEnde - letzte Stelle

Returns:

der Inhalt zwischen den gewählten Positionen der Komponente als Zeichenkette (String)

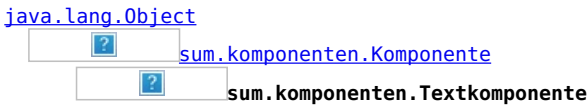
verliertFokus

protected void **verliertFokus**()

Das Textfeld verliert den Fokus.

sum.komponenten

Class Textkomponente



Direct Known Subclasses:
[Etikett](#), [Knopf](#), [Markierungskomponente](#), [Menue](#), [Schalter](#)

```
public abstract class Textkomponente
extends Komponente
```

Die Klasse Textkomponente ist eine abstrakte Oberklasse fuer alle textorientierten Komponenten der SuM-Komponentenbibliothek. Neben den Diensten und Eigenschaften, die von der Oberklasse geerbt werden, besitzt eine Textkomponente einen Inhalt, der in verschiedenen Datentypen abgefragt oder gesetzt werden kann. Standardmaessig ist der Inhalt leer. Das Aussehen des Inhalts kann in Schriftart, Schriftgroesse, Schriftfarbe und Schriftstil veraendert werden.

Version:
7.3 vom 11.10.2011

Author:
Horst Hildebrecht

Field Summary	
protected String	zAktuellFont
protected Font	zSchriftArt
protected int	zSchriftGroesse
protected int	zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Constructor Summary
Textkomponente ()

Method Summary	
int	inhaltAlsGanzeZahl() Der Text der Textkomponente wird als ganze Zahl zurueckgegeben.
long	inhaltAlsLangeGanzeZahl() Der Text der Textkomponente wird als lange ganze Zahl zurueckgegeben.
abstract String	inhaltAlsText() Abstrakter Dienst, der in den Unterklassen ueberschrieben wird.
double	inhaltAlsZahl() Der Text der Textkomponente wird als Kommazahl zurueckgegeben.
boolean	inhaltIstGanzeZahl() Es wird zurueckgegeben, ob die Textkomponente eine ganze Zahl ist.
boolean	inhaltIstLangeGanzeZahl() Es wird zurueckgegeben, ob die Textkomponente eine lange ganze Zahl ist.
boolean	inhaltIstText() Es wird zurueckgegeben, ob die Textkomponente keine (ganze) Zahl ist.

boolean	inhaltIstZahl() Es wird zurueckgegeben, ob die Textkomponente eine Kommazahl ist.
protected void	init (double pLinks, double pOben, double pBreite, double pHoehe, String pInhalt) Position, Breite,Hoehe und Inhalt der Textkomponente werden festgelegt.
Color	schriftfarbe() Die Schriftfarbe der Textkomponente wird zurueckgegeben
Color	schriftFarbe() Die Schriftfarbe der Textkomponente wird zurueckgegeben
void	setzeInhalt (char pZeichen) Die Textkomponente erhaelt einen neuen Inhalt als Zeichen.
void	setzeInhalt (double pZahl) Die Textkomponente erhaelt einen neuen Inhalt als Kommazahl.
void	setzeInhalt (int pZahl) Die Textkomponente erhaelt einen neuen Inhalt als ganze Zahl.
void	setzeInhalt (long pZahl) Die Textkomponente erhaelt einen neuen Inhalt als lange ganze Zahl.
abstract void	setzeInhalt (String pText) Die Textkomponente erhaelt einen neuen Inhalt.
void	setzeSchriftart (String pSchriftart) Die Schriftart der Textkomponente wird veraendert
void	setzeSchriftArt (String pSchriftart) Die Schriftart der Textkomponente wird veraendert
void	setzeSchriftfarbe (Color pFarbe) Die Schriftfarbe der Textkomponente wird veraendert
void	setzeSchriftFarbe (Color pFarbe) Die Schriftfarbe der Textkomponente wird veraendert
void	setzeSchriftfarbe (int pFarbe) Die Schriftfarbe der Textkomponente wird veraendert
void	setzeSchriftFarbe (int pFarbe) Die Schriftfarbe der Textkomponente wird veraendert
void	setzeSchriftgroesse (int pGroesse) Die Schriftgroesse der Textkomponente wird veraendert
void	setzeSchriftGroesse (int pGroesse) Die Schriftgroesse der Textkomponente wird veraendert
void	setzeSchriftstil (int pStil) Der Schriftstil der Textkomponente wird veraendert
void	setzeSchriftStil (int pStil) Der Schriftstil der Textkomponente wird veraendert

Methods inherited from class <code>sum.komponenten.Komponente</code>
aktiviere , besitztFokus , breite , deaktiviere , farbe , fokusErhaltenBearbeiter , fokusVerlorenBearbeiter , gibFrei , hoehe , init , istAktiv , istSichtbar , lerneKomponenteKennen , links , oben , setzeBearbeiterFokusErhalten , setzeBearbeiterFokusVerloren , setzeFarbe , setzeFarbe , setzeFokus , setzeFokusWert , setzeGroesse , setzeHinweis , setzePosition , verstecke , zeige

Methods inherited from class <code>java.lang.Object</code>
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

zAktuellFont

protected [String](#) zAktuellFont

zSchriftArt

protected [Font](#) zSchriftArt

zSchriftGroesse

protected int zSchriftGroesse

zSchriftStil

protected int zSchriftStil

Constructor Detail

Textkomponente

public Textkomponente()

Method Detail

inhaltAlsGanzeZahl

public int inhaltAlsGanzeZahl()
throws [ArithmeticException](#)

Der Text der Textkomponente wird als ganze Zahl zurueckgegeben.

Returns:
der Inhalt der Komponente als ganze Zahl

Throws:
[ArithmeticException](#)

inhaltAlsLangeGanzeZahl

public long inhaltAlsLangeGanzeZahl()
throws [ArithmeticException](#)

Der Text der Textkomponente wird als lange ganze Zahl zurueckgegeben.

Returns:
der Inhalt der Komponente als lange ganze Zahl

Throws:
[ArithmeticException](#)

inhaltAlsText

public abstract [String](#) inhaltAlsText()

Abstrakter Dienst, der in den Unterklassen ueberschrieben wird. In Unterklassen wird der Text der Textkomponente als String zurueckgegeben.

Returns:
der Inhalt der Komponente als Zeichenkette (String)

inhaltAlsZahl

public double inhaltAlsZahl()
throws [ArithmeticException](#)

Der Text der Textkomponente wird als Kommazahl zurueckgegeben.

Returns:
der Inhalt der Komponente als Kommazahl

Throws:
[ArithmeticException](#)

inhaltIstGanzeZahl

public boolean inhaltIstGanzeZahl()

Es wird zurueckgegeben, ob die Textkomponente eine ganze Zahl ist.

Returns:
true, wenn der Inhalt eine ganze Zahl ist

inhaltIstLangeGanzeZahl

```
public boolean inhaltIstLangeGanzeZahl()
```

Es wird zurueckgegeben, ob die Textkomponente eine lange ganze Zahl ist.

Returns:
true, wenn der Inhalt eine lange ganze Zahl ist

inhaltIstText

```
public boolean inhaltIstText()
```

Es wird zurueckgegeben, ob die Textkomponente keine (ganze) Zahl ist.

Returns:
true, wenn der Inhalt keine Zahl ist

inhaltIstZahl

```
public boolean inhaltIstZahl()
```

Es wird zurueckgegeben, ob die Textkomponente eine Kommazahl ist.

Returns:
true, wenn der Inhalt eine Zahl ist

init

```
protected void init(double pLinks,  
                    double pOben,  
                    double pBreite,  
                    double pHoehe,  
                    String pInhalt)
```

Position, Breite,Hoehe und Inhalt der Textkomponente werden festgelegt.

schriftfarbe

```
public Color schriftfarbe()
```

Die Schriftfarbe der Textkomponente wird zurueckgegeben

Returns:
die Schriftfarbe der Textkomponente

schriftFarbe

```
public Color schriftFarbe()
```

Die Schriftfarbe der Textkomponente wird zurueckgegeben

Returns:
die Schriftfarbe der Textkomponente

setzeInhalt

```
public void setzeInhalt(char pZeichen)
```

Die Textkomponente erhaelt einen neuen Inhalt als Zeichen.

Parameters:
pZeichen - das Zeichen, das als neuer Inhalt gesetzt wird

setzeInhalt

```
public void setzeInhalt(double pZahl)
```

Die Textkomponente erhaelt einen neuen Inhalt als Kommazahl.

Parameters:
pZahl - die Zahl, die als neuer Inhalt gesetzt wird

setzeInhalt

```
public void setzeInhalt(int pZahl)
```

Die Textkomponente erhaelt einen neuen Inhalt als ganze Zahl.

Parameters:

pZahl - die Zahl, die als neuer Inhalt gesetzt wird

setzeInhalt

```
public void setzeInhalt(long pZahl)
```

Die Textkomponente erhaelt einen neuen Inhalt als lange ganze Zahl.

Parameters:

pZahl - die Zahl, die als neuer Inhalt gesetzt wird

setzeInhalt

```
public abstract void setzeInhalt(String pText)
```

Die Textkomponente erhaelt einen neuen Inhalt. Leere Methode, wird in der Unterklasse ueberschrieben.

Parameters:

pText - der Text, der als neuer Inhalt gesetzt wird

setzeSchriftart

```
public void setzeSchriftart(String pSchriftart)
```

Die Schriftart der Textkomponente wird veraendert

Parameters:

pSchriftart - die neue Schriftart der Komponente (siehe Klasse Schrift)

setzeSchriftArt

```
public void setzeSchriftArt(String pSchriftart)
```

Die Schriftart der Textkomponente wird veraendert

Parameters:

pSchriftart - die neue Schriftart der Komponente (siehe Klasse Schrift)

setzeSchriftfarbe

```
public void setzeSchriftfarbe(Color pFarbe)
```

Die Schriftfarbe der Textkomponente wird veraendert

Parameters:

pFarbe - die neue Schriftfarbe der Komponente

setzeSchriftFarbe

```
public void setzeSchriftFarbe(Color pFarbe)
```

Die Schriftfarbe der Textkomponente wird veraendert

Parameters:

pFarbe - die neue Schriftfarbe der Komponente

setzeSchriftfarbe

```
public void setzeSchriftfarbe(int pFarbe)
```

Die Schriftfarbe der Textkomponente wird veraendert

Parameters:
pFarbe - die neue Schriftfarbe der Komponente

setzeSchriftFarbe

public void **setzeSchriftFarbe**(int pFarbe)

Die Schriftfarbe der Textkomponente wird veraendert

Parameters:
pFarbe - die neue Schriftfarbe der Komponente

setzeSchriftgroesse

public void **setzeSchriftgroesse**(int pGroesse)

Die Schriftgroesse der Textkomponente wird veraendert

Parameters:
pGroesse - die neue Schriftgroesse der Komponente

setzeSchriftGroesse

public void **setzeSchriftGroesse**(int pGroesse)

Die Schriftgroesse der Textkomponente wird veraendert

Parameters:
pGroesse - die neue Schriftgroesse der Komponente

setzeSchriftstil

public void **setzeSchriftstil**(int pStil)

Der Schriftstil der Textkomponente wird veraendert

Parameters:
pStil - der neue Schriftstil der Komponente (siehe Klasse Schrift)

setzeSchriftStil

public void **setzeSchriftStil**(int pStil)

Der Schriftstil der Textkomponente wird veraendert

Parameters:
pStil - der neue Schriftstil der Komponente (siehe Klasse Schrift)

sum.komponenten

Class Zeichenbereich



All Implemented Interfaces:
[Serializable](#), [ScrollPaneConstants](#)

```
public class Zeichenbereich
extends Textbereich
implements Serializable
```

Ein Zeichenbereich ist die zeichenorientierte Unterklasse des Textbereichs. Zeichenbereiche sind Editoren mit einer festen Breite und Hoehe. Sie reagieren auf Texteingaben und Mausklicks. Wenn etwas in einen Zeichenbereich getippt werden soll, muss es vorher mit einem Mausklick aktiviert werden, d.h. der Zeichenbereich muss den Fokus haben. Der Zeichenbereich ist zeichenorientiert. Fuer zeilenorientierte Bereiche sollte die Klasse Zeilenbereich benutzt werden.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary	
static String	NEUERABSATZ

Fields inherited from class sum.komponenten. Textbereich
hatScrollPane

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Fields inherited from interface javax.swing. ScrollPaneConstants
COLUMN_HEADER , HORIZONTAL_SCROLLBAR , HORIZONTAL_SCROLLBAR_ALWAYS , HORIZONTAL_SCROLLBAR_AS_NEEDED , HORIZONTAL_SCROLLBAR_NEVER , HORIZONTAL_SCROLLBAR_POLICY , LOWER_LEADING_CORNER , LOWER_LEFT_CORNER , LOWER_RIGHT_CORNER , LOWER_TRAILING_CORNER , ROW_HEADER , UPPER_LEADING_CORNER , UPPER_LEFT_CORNER , UPPER_RIGHT_CORNER , UPPER_TRAILING_CORNER , VERTICAL_SCROLLBAR , VERTICAL_SCROLLBAR_ALWAYS , VERTICAL_SCROLLBAR_AS_NEEDED , VERTICAL_SCROLLBAR_NEVER , VERTICAL_SCROLLBAR_POLICY , VIEWPORT

Constructor Summary	
Zeichenbereich (double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Der Zeichenbereich wird erzeugt und enthaelt den uebergebenen Text.
Zeichenbereich (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pText)	Der Zeichenbereich wird erzeugt und enthaelt den uebergebenen Text.

Method Summary	
int anzahl ()	Die Anzahl der Zeichen im Zeichenbereich wird zurueckgegeben.

protected void	bekommtFokus() Der Zeichenbereich erhaelt den Fokus.
void	fuegeEin(String pText, int pStelle) Der Text pText wird an Position pStelle eingefuegt.
void	haengeAn(char pZeichen) Das Zeichen pZeichen wird ans Ende angehaengt.
void	haengeAn(double pZahl) Das Kommazahl pZahl wird am Ende angehaengt.
void	haengeAn(int pZahl) Das ganze Zahl pZahl wird am Ende angehaengt.
void	haengeAn(String pText) Der Text pText wird am Ende angehaengt.
String	inhaltAlsText() Der Text der Zeichenbereich wird als String zurueckgegeben.
protected void	inhaltGeaendert() Der Zeichenbereich reagiert auf Veraenderungen.
boolean	istMarkiert() Es wird zurueckgegeben, ob der Text markiert ist.
void	loesche(int pAnfang, int pEnde) Der durch die Parameter bestimmte Text wird geloesch.
void	loescheAlles() Der gesamte Text wird geloesch.
void	loescheMarkierung() Der markierte Text wird geloesch.
void	markiereAlles() Der gesamte Text wird markiert.
void	markiereNichts() Der gesamte Text wird nicht markiert.
String	markierterInhalt() Der im Textfeld markierte Text wird geliefert.
protected void	markierungGeaendert() Der Zeichenbereich reagiert auf Veraenderungen der Markierung.
int	markierungsAnfang() Es wird zurueckgegeben, von wo an der Text markiert ist.
int	markierungsEnde() Es wird zurueckgegeben, bis wohin der Text markiert ist.
void	neuerAbsatz() Ein neuer Absatz wird erzeugt.
void	setzeBearbeiterInhaltGeaendert(String pBearbeiter) Der Dienst zur Bearbeitung des InhaltGeaendert-Ereignisses in
void	setzeBearbeiterMarkierungGeaendert(String pBearbeiter) Der Dienst zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeInhalt(String pText) Der Zeichenbereich erhaelt einen neuen Inhalt.
void	setzeMarkierung(int pAnfang, int pEnde) Der Text zwischen den Zeichen an Position pAnfang und pEnde wird markiert.
String	teilinhalt(int pAnfang, int pEnde) Der Inhalt des Zeichenbereichs von pAnfang bis pEnde wird als String zurueckgegeben.
protected void	verliertFokus() Der Zeichenbereich verliert den Fokus.

Methods inherited from class **sum.komponenten.Textbereich**

[lerneKomponenteKennen](#), [setzeGroesse](#), [setzePosition](#)

Methods inherited from class **sum.komponenten.Markierungskomponente**

[teilInhalt](#)

Methods inherited from class **sum.komponenten.Textkomponente**

[inhaltAlsGanzeZahl](#), [inhaltAlsLangeGanzeZahl](#), [inhaltAlsZahl](#), [inhaltIstGanzeZahl](#), [inhaltIstLangeGanzeZahl](#), [inhaltIstText](#), [inhaltIstZahl](#), [init](#), [schriftfarbe](#), [schriftFarbe](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeSchriftart](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftgroesse](#),

[setzeSchriftGroesse](#), [setzeSchriftstil](#), [setzeSchriftStil](#)

Methods inherited from class `sum.komponenten.Komponente`

[aktiviere](#), [besitztFokus](#), [breite](#), [deaktiviere](#), [farbe](#), [fokusErhaltenBearbeiter](#), [fokusVerlorenBearbeiter](#), [gibFrei](#), [hoehe](#), [init](#), [istAktiv](#), [istSichtbar](#), [links](#), [oben](#), [setzeBearbeiterFokusErhalten](#), [setzeBearbeiterFokusVerloren](#), [setzeFarbe](#), [setzeFarbe](#), [setzeFokus](#), [setzeFokusWert](#), [setzeHinweis](#), [verstecke](#), [zeige](#)

Methods inherited from class `java.lang.Object`

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

NEUERABSATZ

public static final [String](#) NEUERABSATZ

See Also:

[Constant Field Values](#)

Constructor Detail

Zeichenbereich

```
public Zeichenbereich(double pLinks,
                      double pOben,
                      double pBreite,
                      double pHoehe,
                      String pText)
```

Der Zeichenbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Zeichenbereich befindet sich auf dem Bildschirm.

Parameters:

- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Zeichenbereich

```
public Zeichenbereich(Fenster pFenster,
                      double pLinks,
                      double pOben,
                      double pBreite,
                      double pHoehe,
                      String pText)
```

Der Zeichenbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Zeichenbereich befindet sich auf dem Fenster.

Parameters:

- pFenster - das Fenster, das die Komponente enthaelt
- pLinks - der Abstand der Komponente vom linken Fensterrand
- pOben - der Abstand der Komponente vom oberen Fensterrand
- pBreite - die Breite der Komponente
- pHoehe - die Hoehe der Komponente
- pText - der Inhalt der Komponente

Method Detail

anzahl

```
public int anzahl()
```

Die Anzahl der Zeichen im Zeichenbereich wird zurueckgegeben.

Returns:

die Anzahl der Zeichen der Komponente

bekommtFokus

protected void **bekommtFokus**()

Der Zeichenbereich erhaelt den Fokus.

fuegeEin

public void **fuegeEin**([String](#) pText,
int pStelle)

Der Text pText wird an Position pStelle eingefuegt.

Specified by:
[fuegeEin](#) in class [Textbereich](#)

Parameters:
pText - der Text, der eingefuegt wird
pStelle - die Stelle, wo der Text eingefuegt wird

haengeAn

public void **haengeAn**(char pZeichen)

Das Zeichen pZeichen wird ans Ende angehaengt.

Specified by:
[haengeAn](#) in class [Textbereich](#)

Parameters:
pZeichen - das Zeichen, das ans Ende angehaengt wird

haengeAn

public void **haengeAn**(double pZahl)

Das Kommazahl pZahl wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Textbereich](#)

Parameters:
pZahl - die Zahl, die ans Ende angehaengt wird

haengeAn

public void **haengeAn**(int pZahl)

Das ganze Zahl pZahl wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Textbereich](#)

Parameters:
pZahl - die Zahl, die ans Ende angehaengt wird

haengeAn

public void **haengeAn**([String](#) pText)

Der Text pText wird am Ende angehaengt.

Specified by:
[haengeAn](#) in class [Textbereich](#)

Parameters:
pText - der Text, der ans Ende angehaengt wird

inhaltAlsText

```
public String inhaltAlsText()
```

Der Text der Zeichenbereich wird als String zurueckgegeben.

Specified by:
[inhaltAlsText](#) in class [Textkomponente](#)

Returns:
der Inhalt der Komponente als Zeichenkette (String)

inhaltGeaendert

```
protected void inhaltGeaendert()
```

Der Zeichenbereich reagiert auf Veraenderungen.

istMarkiert

```
public boolean istMarkiert()
```

Es wird zurueckgegeben, ob der Text markiert ist.

Specified by:
[istMarkiert](#) in class [Markierungskomponente](#)

Returns:
true, wenn Text markiert ist

loesche

```
public void loesche(int pAnfang,  
                   int pEnde)
```

Der durch die Parameter bestimmte Text wird geloescht.

Specified by:
[loesche](#) in class [Markierungskomponente](#)

Parameters:
pAnfang - erste Zeichenposition
pEnde - letzte Zeichenposition

loescheAlles

```
public void loescheAlles()
```

Der gesamte Text wird geloescht.

Overrides:
[loescheAlles](#) in class [Markierungskomponente](#)

loescheMarkierung

```
public void loescheMarkierung()
```

Der markierte Text wird geloescht.

Specified by:
[loescheMarkierung](#) in class [Markierungskomponente](#)

markiereAlles

```
public void markiereAlles()
```

Der gesamte Text wird markiert.

Specified by:
[markiereAlles](#) in class [Markierungskomponente](#)

markiereNichts

```
public void markiereNichts()
```

Der gesamte Text wird nicht markiert.

Specified by:
[markiereNichts](#) in class [Markierungskomponente](#)

markierterInhalt

```
public String markierterInhalt()
```

Der im Textfeld markierte Text wird geliefert.

Specified by:
[markierterInhalt](#) in class [Markierungskomponente](#)

Returns:
der markierte Text

markierungGeaendert

```
protected void markierungGeaendert()
```

Der Zeichenbereich reagiert auf Veraenderungen der Markierung.

Specified by:
[markierungGeaendert](#) in class [Markierungskomponente](#)

markierungsAnfang

```
public int markierungsAnfang()
```

Es wird zurueckgegeben, von wo an der Text markiert ist.

Specified by:
[markierungsAnfang](#) in class [Markierungskomponente](#)

Returns:
die erste markierte Position

markierungsEnde

```
public int markierungsEnde()
```

Es wird zurueckgegeben, bis wohin der Text markiert ist.

Specified by:
[markierungsEnde](#) in class [Markierungskomponente](#)

Returns:
die letzte markierte Position

neuerAbsatz

```
public void neuerAbsatz()
```

Ein neuer Absatz wird erzeugt.

setzeBearbeiterInhaltGeaendert

```
public void setzeBearbeiterInhaltGeaendert(String pBearbeiter)
```

Der Dienst zur Bearbeitung des InhaltGeaendert-Ereignisses in

Overrides:
[setzeBearbeiterInhaltGeaendert](#) in class [Markierungskomponente](#)

Parameters:
pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Inhalt der Komponente geaendert wurde. der Ereignisanwendung wird festgelegt.

setzeBearbeiterMarkierungGeaendert

public void **setzeBearbeiterMarkierungGeaendert**([String](#) pBearbeiter)

Der Dienst zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Overrides:

[setzeBearbeiterMarkierungGeaendert](#) in class [Markierungskomponente](#)

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Markierung der Komponente geaendert wurde.

setzeInhalt

public void **setzeInhalt**([String](#) pText)

Der Zeichenbereich erhaelt einen neuen Inhalt.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der Text, der als neuer Inhalt gesetzt wird

setzeMarkierung

public void **setzeMarkierung**(int pAnfang,
int pEnde)

Der Text zwischen den Zeichen an Position pAnfang und pEnde wird markiert.

Specified by:

[setzeMarkierung](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Zeichenposition
pEnde - letzte Zeichenposition

teihinhalt

public [String](#) **teihinhalt**(int pAnfang,
int pEnde)

Der Inhalt des Zeichenbereichs von pAnfang bis pEnde wird als String zurueckgegeben.

Specified by:

[teihinhalt](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Zeichenposition
pEnde - letzte Zeichenposition

Returns:

der Inhalt in den gewaehlten Zeilen der Komponente als Zeichenkette (String)

verliertFokus

protected void **verliertFokus**()

Der Zeichenbereich verliert den Fokus.

sum.komponenten

Class Zeilenbereich



All Implemented Interfaces:
[Serializable](#), [ScrollPaneConstants](#)

```
public class Zeilenbereich
extends Textbereich
implements Serializable
```

Ein Zeilenbereich ist die zeilenorientierte Unterklasse des Textbereichs. Zeilenbereiche sind zeilenorientierte Editoren mit einer festen Breite und Hoehe. Sie reagieren auf Texteingaben und Mausklicks. Wenn etwas in einen Zeilenbereich getippt werden soll, muss er vorher mit einem Mausklick aktiviert werden, d.h. der Zeilenbereich muss den Fokus haben. Der Zeilenbereich ist zeilenorientiert. Fuer zeichenorientierte Bereiche sollte die Klasse Zeilenbereich benutzt werden.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek, Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary	
static String	NEUERABSATZ

Fields inherited from class sum.komponenten. Textbereich
hatScrollPane

Fields inherited from class sum.komponenten. Textkomponente
zAktuellFont , zSchriftArt , zSchriftGroesse , zSchriftStil

Fields inherited from class sum.komponenten. Komponente
hatComponent , kenntFenster

Fields inherited from interface javax.swing. ScrollPaneConstants
COLUMN_HEADER , HORIZONTAL_SCROLLBAR , HORIZONTAL_SCROLLBAR_ALWAYS , HORIZONTAL_SCROLLBAR_AS_NEEDED , HORIZONTAL_SCROLLBAR_NEVER , HORIZONTAL_SCROLLBAR_POLICY , LOWER_LEADING_CORNER , LOWER_LEFT_CORNER , LOWER_LEADING_CORNER , LOWER_RIGHT_CORNER , LOWER_TRAILING_CORNER , ROW_HEADER , UPPER_LEADING_CORNER , UPPER_LEFT_CORNER , UPPER_RIGHT_CORNER , UPPER_TRAILING_CORNER , VERTICAL_SCROLLBAR , VERTICAL_SCROLLBAR_ALWAYS , VERTICAL_SCROLLBAR_AS_NEEDED , VERTICAL_SCROLLBAR_NEVER , VERTICAL_SCROLLBAR_POLICY , VIEWPORT

Constructor Summary
Zeilenbereich (double pLinks, double pOben, double pBreite, double pHoehe, String pText) Der Zeilenbereich wird erzeugt und enthaelt den uebergebenen Text.
Zeilenbereich (Fenster pFenster, double pLinks, double pOben, double pBreite, double pHoehe, String pText) Der Zeilenbereich wird erzeugt und enthaelt den uebergebenen Text.

Method Summary
int

	anzahl() Die Anzahl der Zeilen im Zeilenbereich wird zurueckgegeben.
protected void	bekommtFokus() Der Zeilenbereich erhaelt den Fokus.
void	fuegeEin(String pText, int pZeile) Der Text pText wird als Zeile pZeile eingefuegt.
void	haengeAn(char pZeichen) Das Zeichen pZeichen wird als neue Zeile am Ende angehaengt.
void	haengeAn(double pZahl) Das Kommazahl pZahl wird als neue Zeile am Ende angehaengt.
void	haengeAn(int pZahl) Das ganze Zahl pZahl wird als neue Zeile am Ende angehaengt.
void	haengeAn(String pText) Der Text pText wird als neue Zeile am Ende angehaengt.
String	inhaltAlsText() Der Text des Zeilenbereichs wird als String zurueckgegeben.
protected void	inhaltGeaendert() Der Zeilenbereich reagiert auf Veraenderungen.
boolean	istMarkiert() Es wird zurueckgegeben, ob Text markiert ist.
void	loesche(int pAnfang, int pEnde) Der durch die Parameter bestimmte Text wird geloescht.
void	loescheAlles() Der gesamte Text wird geloescht.
void	loescheMarkierung() Der markierte Text wird geloescht.
void	markiereAlles() Der gesamte Text wird markiert.
void	markiereNichts() Der gesamte Text wird nicht markiert.
String	markierterInhalt() Der im Zeilenbereich markierte Text wird geliefert.
protected void	markierungGeaendert() Der Zeilenbereich reagiert auf Veraenderungen der Markierung.
int	markierungsAnfang() Es wird zurueckgegeben, von welcher Zeile an der Text markiert ist.
int	markierungsEnde() Es wird zurueckgegeben, bis zu welcher Zeile der Text markiert ist.
void	neuerAbsatz() Ein neuer Absatz wird erzeugt.
void	setzeBearbeiterInhaltGeaendert(String pBearbeiter) Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeBearbeiterMarkierungGeaendert(String pBearbeiter) Die Methode zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.
void	setzeInhalt(String pText) Der Zeilenbereich erhaelt einen neuen Inhalt.
void	setzeMarkierung(int pAnfang, int pEnde) Der Text zwischen den Zeilen pAnfang und pEnde wird markiert.
String	teilinhalt(int pAnfang, int pEnde) Der Inhalt des Zeilenbereichs von Zeile pAnfang bis Zeile pEnde wird als String zurueckgegeben.
protected void	verliertFokus() Der Zeilenbereich verliert den Fokus.

Methods inherited from class **sum.komponenten.Textbereich**

[lerneKomponenteKennen](#), [setzeGroesse](#), [setzePosition](#)

Methods inherited from class **sum.komponenten.Markierungskomponente**

[teilinhalt](#)

Methods inherited from class sum.komponenten.[Textkomponente](#)

[inhaltAlsGanzeZahl](#), [inhaltAlsLangeGanzeZahl](#), [inhaltAlsZahl](#), [inhaltIstGanzeZahl](#), [inhaltIstLangeGanzeZahl](#), [inhaltIstText](#), [inhaltIstZahl](#), [init](#), [schriftfarbe](#), [schriftFarbe](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeInhalt](#), [setzeSchriftart](#), [setzeSchriftArt](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftfarbe](#), [setzeSchriftFarbe](#), [setzeSchriftFarbe](#), [setzeSchriftgroesse](#), [setzeSchriftGroesse](#), [setzeSchriftstil](#), [setzeSchriftStil](#)

Methods inherited from class sum.komponenten.[Komponente](#)

[aktiviere](#), [besitztFokus](#), [breite](#), [deaktiviere](#), [farbe](#), [fokusErhaltenBearbeiter](#), [fokusVerlorenBearbeiter](#), [gibFrei](#), [hoehe](#), [init](#), [istAktiv](#), [istSichtbar](#), [links](#), [oben](#), [setzeBearbeiterFokusErhalten](#), [setzeBearbeiterFokusVerloren](#), [setzeFarbe](#), [setzeFarbe](#), [setzeFokus](#), [setzeFokusWert](#), [setzeHinweis](#), [verstecke](#), [zeige](#)

Methods inherited from class java.lang.[Object](#)

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Field Detail

NEUERABSATZ

public static final [String](#) NEUERABSATZ

See Also:
[Constant Field Values](#)

Constructor Detail

Zeilenbereich

```
public Zeilenbereich(double pLinks,
                    double pOben,
                    double pBreite,
                    double pHoehe,
                    String pText)
```

Der Zeilenbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Zeilenbereich befindet sich auf dem Bildschirm.

- Parameters:**
- pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pText - der Inhalt der Komponente

Zeilenbereich

```
public Zeilenbereich(Fenster pFenster,
                    double pLinks,
                    double pOben,
                    double pBreite,
                    double pHoehe,
                    String pText)
```

Der Zeilenbereich wird erzeugt und enthaelt den uebergebenen Text. Position, Breite und Hoehe werden als Parameter uebergeben. Der Zeilenbereich befindet sich auf dem Fenster.

- Parameters:**
- pFenster - das Fenster, das die Komponente enthaelt
 - pLinks - der Abstand der Komponente vom linken Fensterrand
 - pOben - der Abstand der Komponente vom oberen Fensterrand
 - pBreite - die Breite der Komponente
 - pHoehe - die Hoehe der Komponente
 - pText - der Inhalt der Komponente

Method Detail

anzahl

```
public int anzahl()
```

Die Anzahl der Zeilen im Zeilenbereich wird zurueckgegeben.

Returns:

die Anzahl der Zeilen der Komponente

bekommtFokus

```
protected void bekommtFokus()
```

Der Zeilenbereich erhaelt den Fokus.

fuegeEin

```
public void fuegeEin(String pText,  
                    int pZeile)
```

Der Text pText wird als Zeile pZeile eingefuegt.

Specified by:

[fuegeEin](#) in class [Textbereich](#)

Parameters:

pText - der Text, der eingefuegt wird

pZeile - die Zeilennummer, wo der Text als neue Zeile eingefuegt wird

haengeAn

```
public void haengeAn(char pZeichen)
```

Das Zeichen pZeichen wird als neue Zeile am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Textbereich](#)

Parameters:

pZeichen - das Zeichen, das als neue Zeile angehaengt wird

haengeAn

```
public void haengeAn(double pZahl)
```

Das Kommazahl pZahl wird als neue Zeile am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Textbereich](#)

Parameters:

pZahl - die Zahl, die als neue Zeile angehaengt wird

haengeAn

```
public void haengeAn(int pZahl)
```

Das ganze Zahl pZahl wird als neue Zeile am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Textbereich](#)

Parameters:

pZahl - die Zahl, die als neue Zeile angehaengt wird

haengeAn

```
public void haengeAn(String pText)
```

Der Text pText wird als neue Zeile am Ende angehaengt.

Specified by:

[haengeAn](#) in class [Textbereich](#)

Parameters:

pText - der Text, der als neue Zeile angehaengt wird

inhaltAlsText

public [String](#) **inhaltAlsText**()

Der Text des Zeilenbereichs wird als String zurueckgegeben.

Specified by:

[inhaltAlsText](#) in class [Textkomponente](#)

Returns:

der Inhalt der Komponente als Zeichenkette (String)

inhaltGeaendert

protected void **inhaltGeaendert**()

Der Zeilenbereich reagiert auf Veraenderungen.

istMarkiert

public boolean **istMarkiert**()

Es wird zurueckgegeben, ob Text markiert ist.

Specified by:

[istMarkiert](#) in class [Markierungskomponente](#)

Returns:

true, wenn Text markiert ist

loesche

public void **loesche**(int pAnfang,
int pEnde)

Der durch die Parameter bestimmte Text wird geloescht. Die Zeilennummerierung beginnt bei 1.

Specified by:

[loesche](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Zeile
pEnde - letzte Zeile

loescheAlles

public void **loescheAlles**()

Der gesamte Text wird geloescht.

Overrides:

[loescheAlles](#) in class [Markierungskomponente](#)

loescheMarkierung

public void **loescheMarkierung**()

Der markierte Text wird geloescht.

Specified by:

[loescheMarkierung](#) in class [Markierungskomponente](#)

markiereAlles

public void **markiereAlles**()

Der gesamte Text wird markiert.

Specified by:

[markiereAlles](#) in class [Markierungskomponente](#)

markiereNichts

```
public void markiereNichts()
```

Der gesamte Text wird nicht markiert.

Specified by:

[markiereNichts](#) in class [Markierungskomponente](#)

markierterInhalt

```
public String markierterInhalt()
```

Der im Zeilenbereich markierte Text wird geliefert.

Specified by:

[markierterInhalt](#) in class [Markierungskomponente](#)

Returns:

der markierte Text

markierungGeaendert

```
protected void markierungGeaendert()
```

Der Zeilenbereich reagiert auf Veraenderungen der Markierung.

Specified by:

[markierungGeaendert](#) in class [Markierungskomponente](#)

markierungsAnfang

```
public int markierungsAnfang()
```

Es wird zurueckgegeben, von welcher Zeile an der Text markiert ist. Die Zeilennummerierung beginnt bei 1.

Specified by:

[markierungsAnfang](#) in class [Markierungskomponente](#)

Returns:

die erste markierte Zeile

markierungsEnde

```
public int markierungsEnde()
```

Es wird zurueckgegeben, bis zu welcher Zeile der Text markiert ist. Die Zeilennummerierung beginnt bei 1.

Specified by:

[markierungsEnde](#) in class [Markierungskomponente](#)

Returns:

die letzte markierte Zeile

neuerAbsatz

```
public void neuerAbsatz()
```

Ein neuer Absatz wird erzeugt.

setzeBearbeiterInhaltGeaendert

```
public void setzeBearbeiterInhaltGeaendert(String pBearbeiter)
```

Die Methode zur Bearbeitung des InhaltGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Overrides:

[setzeBearbeiterInhaltGeaendert](#) in class [Markierungskomponente](#)

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn der Inhalt der Komponente geaendert wurde.

setzeBearbeiterMarkierungGeaendert

public void **setzeBearbeiterMarkierungGeaendert**([String](#) pBearbeiter)

Die Methode zur Bearbeitung des MarkierungGeaendert-Ereignisses in der Ereignisanwendung wird festgelegt.

Overrides:

[setzeBearbeiterMarkierungGeaendert](#) in class [Markierungskomponente](#)

Parameters:

pBearbeiter - der Bezeichner des Dienstes der SuMANwendung, der aufgerufen wird, wenn die Markierung der Komponente geaendert wurde.

setzeInhalt

public void **setzeInhalt**([String](#) pText)

Der Zeilenbereich erhaelt einen neuen Inhalt.

Specified by:

[setzeInhalt](#) in class [Textkomponente](#)

Parameters:

pText - der Text, der als neuer Inhalt gesetzt wird

setzeMarkierung

public void **setzeMarkierung**(int pAnfang,
int pEnde)

Der Text zwischen den Zeilen pAnfang und pEnde wird markiert. Die Zeilennummerierung beginnt bei 1.

Specified by:

[setzeMarkierung](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Zeile
pEnde - letzte Zeile

teillinhalt

public [String](#) **teillinhalt**(int pAnfang,
int pEnde)

Der Inhalt des Zeilenbereichs von Zeile pAnfang bis Zeile pEnde wird als String zurueckgegeben. Die Zeilennummerierung beginnt bei 1.

Specified by:

[teillinhalt](#) in class [Markierungskomponente](#)

Parameters:

pAnfang - erste Zeile
pEnde - letzte Zeile

Returns:

der Inhalt in den gewaehlten Zeilen der Komponente als Zeichenkette (String)

verliertFokus

protected void **verliertFokus**()

Der Zeilenbereich verliert den Fokus.

