



Javaゼミ第15回目

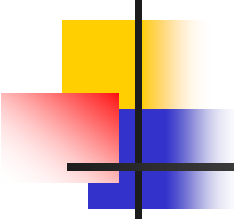
15.8 テキストフィールドとテキストエリア
15.9 リスト

発表日 7月25日 発表者 加藤 友宏



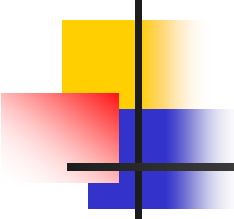
テキストフィールドとテキストエリア

- テキストの入力を可能にする
 - テキストフィールド…1行
 - テキストエリア…複数行
- 文字を入力 テキストイベント生成
- テキストフィールドでは、Enterキーを押したときアクションイベントも生成
- TextFieldとTextAreaは、TextComponentを拡張したもの



TextComponentクラスの主なインスタンスメソッド(1)

メソッド	説明
<code>void addTextListener(TextListener tl)</code>	このコンポーネントで生成されたテキストイベントを受け取るためにtlを登録する
<code>int getCaretPosition()</code>	現在の文字挿入箇所を表すキャレットの位置を返す
<code>String getSelectedText()</code>	ユーザーが入力したテキストから成る文字列を返す
<code>int getSelectionEnd()</code>	選択されたテキストの最後の文字のインデックスを返す
<code>int getSelectionStart()</code>	選択されたテキストの最初の文字のインデックスを返す
<code>String getText()</code>	このコンポーネントのテキスト内容を持つ文字列を返す
<code>boolean isEditable()</code>	コンポーネントが編集可能な場合には真、編集可能ではない場合は偽を返す
<code>void removeTextListener(TextListener tl)</code>	このコンポーネントで生成されたテキストイベントを受け取るためのtlの登録を解除する



TextComponentクラスの主なインスタンスメソッド(2)

メソッド	説明
void select(int start, int end)	インデックスstartからインデックスendまでのテキストを選択する
void selectAll()	このコンポーネント内のすべてのテキストを選択する
void setCaretPosition(int p)	キャレットの位置をpに設定する
void setEditable(boolean flag)	flagが真の場合にはコンポーネントは編集可能、偽の場合は編集できない
void setSelectionEnd(int end)	選択テキストの最後をインデックスendとする
void setSelectionStart(int start)	選択テキストの最初をインデックスstartとする
void setText(String str)	コンポーネントのテキストをstrに設定する



TextFieldクラスのコンストラクタ

TextField()

TextField(String str)

TextField(int cols)

TextField(String str, int cols)

- `str` はフィールドに入力されたテキスト
- `cols` は、フィールドの幅を文字数で指定

TextFieldクラスのインスタンスメソッド

メソッド	説明
void addActionListener(ActionListener al)	このコンポーネントで生成されたアクションイベントを受け取るためにalを登録する
char getEchoChar()	このコンポーネントのエコー文字を返す
void setEchoChar(char c)	このコンポーネントのエコー文字を設定する
void removeActionListener(ActionListener al)	このコンポーネントで生成されたアクションを受け取るためのalを解除する



TextAreaクラスのコンストラクタ(1)

TextArea()

TextArea(String str)

TextArea(int rows, int cols)

TextArea(String str, int rows, int cols)

TextArea(String str, int rows, int cols, int
scrollbars)

- str・・・入力されたテキスト
- 引数rows,cols・・・行と列の数



TextAreaクラスのコンストラクタ(2)

- 引数scrollbars・・・水平、垂直または両方のスクロールバーの作成を指定
- 有効な値は、以下の4つ

値	意味
SCROLLBARS_BOTH	水平と垂直の両方のスクロールバーの作成
SCROLLBARS_HORIZONTAL_ONLY	水平のスクロールバーのみを作成
SCROLLBARS_VERTICAL_ONLY	垂直のスクロールバーのみを作成
SCROLLBARS_NONE	スクロールバーを作成しない



TextAreaクラスのメソッド(1)

- コンポーネントの最後に
文字列strを追加する

```
void append(String str)
```

- 位置indexの文字の直前に
文字列strを挿入する

```
void insert(String str, int index)
```



TextAreaクラスのメソッド(2)

- 位置startと位置endの間の文字範囲を文字列strに書き換える

```
void replaceRange  
(String str, int start, int end)
```



TextEventクラスのコンストラクタ

```
TextEvent(Object src, int type)
```

- src・・・イベントを生成したオブジェクト
- type・・・コマンドのタイプ

TextEventオブジェクトには入力した文字列は含まれない

TextArea, TextFieldのメソッドで文字を読み取る



TextListenerについて

- テキストリスナが実装しなければならないメソッド

```
void textValueChanged(TextEvent te)
```

- te・・・テキストイベント



テキストフィールドの使用手順

1. TextFieldオブジェクトを作成
2. テキストフィールドで生成されたアクションイベント、テキストイベントまたは両方を受け取るためにアプレットを登録
3. add()メソッドを呼び出し、アプレットにTextFieldオブジェクトの追加
4. アプレットにActionListenerとTextListenerの登録



テキストフィールドとテキストエリアの例(1)

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*
  <applet code="TextFieldEvents"
    width=400 height=200>
  </applet>
*/
```

```
public class Swich extends Applet
implements ActionListener, TextListener
{
  TextArea ta;
  TextField tf;
```

```
public void init() {
  tf = new TextField(20);
  tf.addActionListener(this);
  tf.addTextListener(this);
  ta = new TextArea(10, 20);
  add(tf);
  add(ta);
}
```

テキストフィールドとテキストエリアの例(2)

```
public void actionPerformed(ActionEvent ae) {  
    ta.append("ActionEvent: " +  
        ae.getActionCommand() + "\n");  
    tf.setText("");  
}
```

```
public void textValueChanged(TextEvent te) {  
    ta.append("TextEvent: " +  
        tf.getText() + "\n");  
}  
}
```

実行結果





リスト

- 1つ以上の項目を選択するときに使う
- リストが大きすぎて、表示しきれないときは、スクロールバーが自動的に生成される
- リスト内の項目をダブルクリックするとアクションイベントが生成される
- 選択または選択解除すると、項目イベントが生成される



リストのコンストラクタ

List()

List(int rows)

List(int rows, boolean multiple)

- rows・・・表示可能な項目数
- multiple・・・真なら複数選択、偽なら1つの項目



リストの使用手順

1. Listオブジェクトの作成
2. リストに項目の追加
3. アクションイベント、項目イベントまたはその両方を受け取るためにアプレットを登録
4. add()メソッドを呼び出して、アプレットにListオブジェクトを追加
5. アプレットにActionListenerとItemListenerの実装



Listクラスのインスタンスメソッド(1)

メソッド	説明
<code>void add(String str)</code>	strを項目として追加する
<code>void add(String str, int i)</code>	インデックスiにstrを項目として追加する
<code>void addActionListener(ActionListener al)</code>	このリストで生成された項目イベントを受け取るためにalを登録する
<code>void addItemListener(ItemListener il)</code>	このリストで生成された項目イベントを受け取るためにilを登録する
<code>void deselect(int i)</code>	インデックスiの項目の選択を解除する
<code>String getItem(int i)</code>	インデックスiの項目を返す
<code>int getItemCount()</code>	項目の総数を返す



Listクラスのインスタンスメソッド(2)

メソッド	説明
String[] getItems()	すべての項目を含む文字列の配列を返す
int getSelectedIndex()	選択された項目のインデックスを返す
int[] getSelectedindexs()	選択された項目のインデックスを含むint型の配列を返す
String getSelectedItem()	選択された項目を返す
String[] getSelectedItems()	選択された項目を含む文字列の配列を返す
Object[] getSelectedObjects()	選択された項目を含むオブジェクトの配列を返す
boolean isMultipleMode()	複数の項目を選択できる場合は真、選択できない場合は偽を返す

Listクラスのインスタンスメソッド(3)

メソッド	説明
<code>void remove(String str) throws IllegalArgumentException</code>	リストからstrを除去する
<code>void remove(int i)</code>	インデックスiの項目を除去する
<code>void removeAll()</code>	すべての項目を除去する
<code>void removeActionListener(ActionListener al)</code>	このリストで生成されたアクションイベントを受け取るためのalの登録を解除する
<code>void removeItemListener(ItemListener il)</code>	このリストで生成された項目イベントを受け取るためのilの登録を解除する
<code>void select(int i)</code>	インデックスiの項目を選択する
<code>void setMultipleMode(boolean flag)</code>	flagが真の場合には複数の項目を選択することができ、偽の場合には1つの項目しか選択できない



リストの例

リストのアクションイベントと項目イベントを使用した例

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*
  <applet code="ListEvents" width=400 height=200>
  </applet>
*/

public class Sample1 extends Applet
implements ActionListener, ItemListener {
    TextArea ta;
```

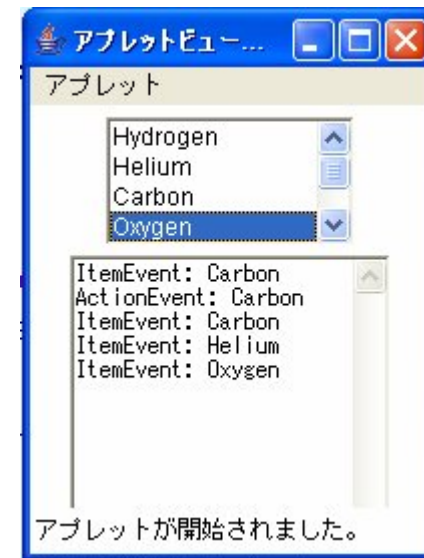
```
public void init() {
    List list = new List();
    list.add("Hydrogen");
    list.add("Helium");
    list.add("Carbon");
    list.add("Oxygen");
    list.add("Potassium");
    list.add("Phosphorus");
    list.addActionListener(this);
    list.addItemListener(this);
    add(list);
    ta = new TextArea(10, 20);
    add(ta);
}
```

リストの例

```
public void actionPerformed(ActionEvent ae) {  
    ta.append("ActionEvent: " +  
        ae.getActionCommand() + "\n");  
}
```

```
public void itemStateChanged(ItemEvent ie) {  
    List list = (List)ie.getItemSelectable();  
    ta.append("ItemEvent: " +  
        list.getSelectedItem() + "\n");  
}
```

実行結果





練習問題

- テキストフィールドに数字を入力し、リストにある四則演算の種類を選ぶと演算結果がテキストエリアに表示されるプログラムを作成せよ。
- テキストフィールドに数字を入力し、Enterキーを押した後にリストの項目を選択することにより演算結果が表示されるようにすること。