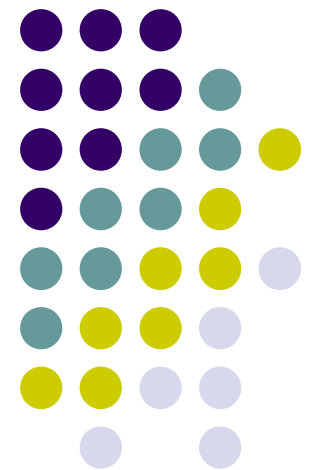
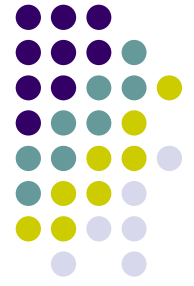


第15回 独習Javaゼミ

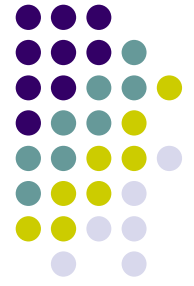
- ・15.5 チェックボックス
- ・15.6 チェックボックスグループ
- ・15.7 チョイス





チェックボックスの概要

- ラベルと小さいボックスから成るコンポーネント
- 状態に応じてボックス内のチェックマークの表示、非表示が決まる
- 状態はボックス上でのマウスのクリックによって変化する
- 用途として、オプションの選択などに使われることが多い



コンストラクタ

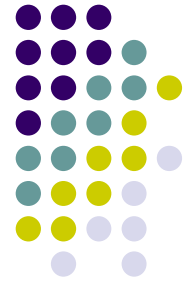
- Checkbox()
- Checkbox(String str)
- Checkbox(String str, boolean state)
- Checkbox(String str, boolean state, CheckboxGroup grp)
- Checkbox(String str, CheckboxGroup grp, boolean state)

str:テキスト

state:チェックボックスの初期値

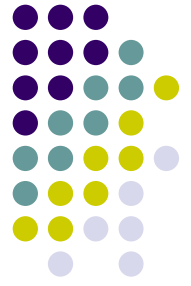
 true: チェックマークが表示 false: マークが非表示

grp:チェックボックスグループの参照



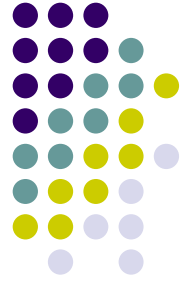
メソッド(1/3)

- チェックボックスの状態が変化すると、項目イベントが生成される
- それらのイベントを受け取るための、登録および登録解除を行うメソッド
- `void addItemListener (ItemListener il)`
- `void removeItemListener (ItemListener il)`
il:項目リスナ



メソッド(2/3)

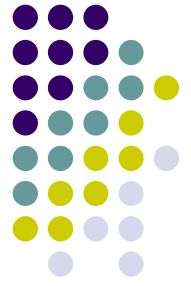
- チェックボックスの状態の読み取りと書き込み
 - `boolean getState()`
 - `void setState (boolean value)`
value: 真:マークが表示 偽:クリア
- ラベルの読み取りと書き込み
 - `String getLabel()`
 - `void setLabel (String str)`
str:ラベルとして使用する文字列



メソッド(3/3)

- チェックボックスのグループを管理する
 - CheckboxGroup getCheckboxGroup()
 - Void setCheckboxGroup (CheckboxGroup grp)
grp:チェックボックスが属するグループ

java.awt.event.ItemEventクラス



- 項目イベントに変化があったときに、それをパラメータとして受け取るItemEventクラスが存在する
- 項目イベントのタイプを識別する2つの定数、DESELECTEDおよびSELECTEDが定義される

* コンストラクタ

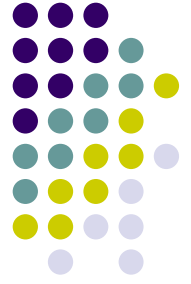
ItemEvent (ItemSelectable src, int type, Object item, int state)

src: イベントを生成したオブジェクト

type: イベントのタイプ

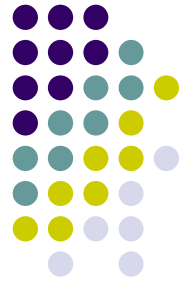
item: イベントを生成した項目

state: イベントの状態



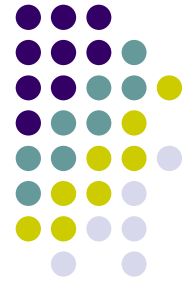
メソッド

- イベントを生成した項目の取得
 - Object getItem()
- イベントを生成したコンポーネントの取得
 - ItemSelectable getItemSelectable()
- 項目の状態(DESELECTED,SELECTED)の取得
 - int getStateChange()



ItemSelectableインターフェイス

- Java.awt.ItemSelectableインターフェイスは、Checkbox, Choice, List, CheckboxMenuItemの各クラスで実装
- 登録・解除メソッド
 - void addItemListener (ItemListener il) il:項目リスナ
 - void removeItemListener (ItemListener il)
- オブジェクトの参照の配列
 - Object[] getSelectedObjects()



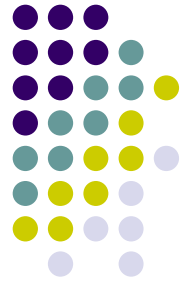
ItemListenerインターフェイス

- java.awt.event.ItemListenerインターフェイスには、項目リスナで実装しなければならないメソッドが存在する
- 項目イベントに変化があったとき、実行する処理を定義する
 - void itemStateChanged (ItemEvent ie)
ie:項目イベント



チェックボックスの使用手順

- 1 . Checkboxオブジェクトを作成する
- 2 . チェックボックスで生成されたイベントを受け取るためにアプレットを登録する
- 3 . add()メソッドを呼び出して、アプレットにCheckboxオブジェクトを追加する
- 4 . ItemListenerインターフェイスを実装する



使用例(1/2)

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="CB" width=400
  height=60></applet>*/
```

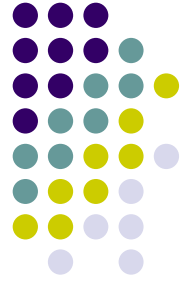
```
public class CB extends Applet
  implements ItemListener{
  Label label;
```

```
  public void init(){
    Checkbox cb1 = new
    Checkbox("Apple");
    cb1.addItemListener(this);
    add(cb1);
```

```
    Checkbox cb2 = new
      Checkbox("Banana");
    cb2.addItemListener(this);
    add(cb2);
```

```
    Checkbox cb3 = new
      Checkbox("Orange");
    cb3.addItemListener(this);
    add(cb3);
```

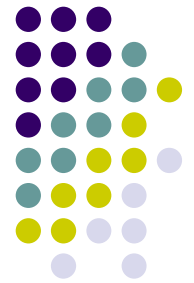
```
    label = new Label("
");
    add(label);
  }
```

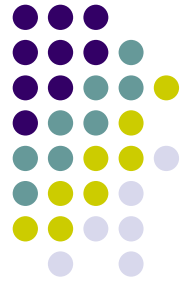


使用例(2/2)

```
public void itemStateChanged(ItemEvent ie){  
    Checkbox cb =  
    (Checkbox)ie.getItemSelectable();  
    label.setText(cb.getLabel() +" "+  
    cb.getState());  
}  
}
```

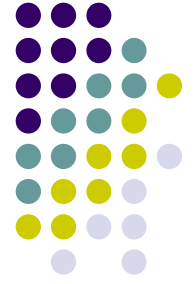
実行結果





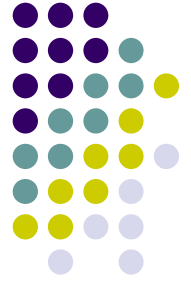
チェックボックスグループ

- 常に1つしか設定できない、チェックボックスのセットのこと(つまりラジオボタン)
- 1つの要素をチェックすると、同じグループのその他の要素は自動的にクリアされる
- 状態を変更すると項目イベントが発生
- 複数のオプションから1つだけを選択させるような場合によく使用される



コンストラクタとメソッド

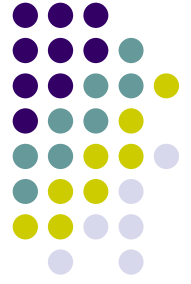
- CheckboxGroup()
 - 現在設定されているグループメンバを取得するメソッド
 - Checkbox getSelectedCheckbox
 - グループ内の特定のチェックボックスを設定するメソッド
 - void setSelectedCheckbox (Checkbox cb)
cb:設定するチェックボックス



使用手順

- 1 . CheckboxGroupオブジェクトを作成
- 2 . Checkboxオブジェクトを作成し、CheckboxGroupオブジェクトの参照をCheckboxコンストラクタに渡す
- 3 . チェックボックスで生成された項目イベント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Checkboxオブジェクトを追加
- 5 . グループ内の各チェックボックスごとに2 ~ 4を繰り返す
- 6 . アプレットにItemListenerインターフェイスを実装する

使用例(1/2)

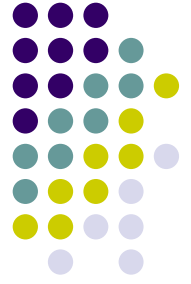


```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="CBG" width=400
  height=60></applet>
*/

public class CBG extends Applet
  implements ItemListener{
  Label label;

  public void init(){
    CheckboxGroup cbg = new
      CheckboxGroup();
```

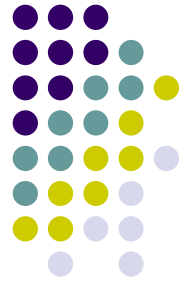
```
    Checkbox cb1 = new
      Checkbox("Apple", cbg, true);
    cb1.addItemListener(this);
    add(cb1);
    Checkbox cb2 = new
      Checkbox("Banana", cbg, false);
    cb2.addItemListener(this);
    add(cb2);
    Checkbox cb3 = new
      Checkbox("Orange", cbg, false);
    cb3.addItemListener(this);
    add(cb3);
    label = new Label("      ");
    add(label);
  }
```

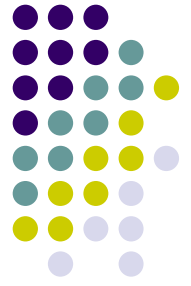


使用例(2/2)

```
public void itemStateChanged(ItemEvent ie){  
    Checkbox cb =  
        (Checkbox)ie.getItemSelectable();  
    label.setText(cb.getLabel() +" "+  
        cb.getState());  
}  
}
```

実行結果



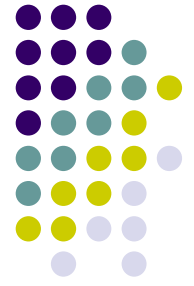


チョイス

- プルダウンメニューのこと
- チョイスをクリックすると有効な選択項目がすべて表示される
- 選択項目の1つを選択すると、チョイスは通常の状態に戻り、現在の選択項目だけが表示される
- 最小限の表示スペースしか必要としないため、非常に重宝される

* コンストラクタ

Choice()



使用手順

- 1 . Choiceオブジェクトを作成する
- 2 . チョイスに項目を追加する
- 3 . チョイスで生成された項目イベントを受け取るために、アプレットを登録する
- 4 . add()メソッドを呼び出して、アプレットにChoiceオブジェクトを追加する
- 5 . アプレットにItemListenerインターフェイスを実装する



インスタンスメソッド(1/2)

- Choiceクラスに定義される主なインスタンスメソッド

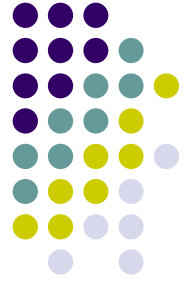
| メソッド | 説明 |
|---|-----------------------------------|
| <code>void add(String str) throws NullPointerException</code> | 項目としてstrを追加する |
| <code>void addItem(String str) throws NullPointerException</code> | 項目としてstrを追加する |
| <code>void addItemListener(ItemListener il)</code> | このチョイスで生成された項目イベントを受け取るためにilを登録する |
| <code>String getItem(int i)</code> | このチョイス内のインデックスiの文字列を返す |
| <code>int getSelectedIndex()</code> | 選択された項目のインデックスを返す |
| <code>String getSelectedItem()</code> | 選択された項目を返す |



インスタンスメソッド(2/2)

| | |
|---|--------------------------------------|
| <code>Object[] getSelectedObjects()</code> | 選択された項目である、1つのオブジェクトを含む配列を返す |
| <code>void insert(String str, int i) throws IllegalArgumentException</code> | インデックス <i>i</i> の前にstrを挿入する |
| <code>void remove(int i) throws IllegalArgumentException</code> | インデックス <i>i</i> の項目を削除する |
| <code>void removeAll()</code> | すべての項目を削除する |
| <code>void removeItemListener(ItemListener il)</code> | このチョイス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) throws IllegalArgumentException</code> | インデックス <i>i</i> の項目を選択する |
| <code>void select(String str) throws IllegalArgumentException</code> | 項目strを選択する |

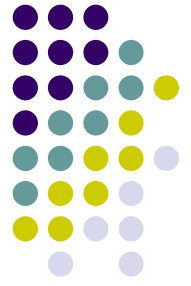
使用例(1/2)



```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
/*<applet code="CHOICE"
   width=400
   height=60></applet>
*/

public class CHOICE extends
    Applet implements
    ItemListener{
    Label label;
```

```
public void init() {
    Choice c1 = new Choice();
    c1.addItem("Red");
    c1.addItem("Orange");
    c1.addItem("Yellow");
    c1.addItem("Green");
    c1.addItem("Blue");
    c1.addItem("Indigo");
    c1.addItem("Violet");
    c1.addItemListener(this);
    add(c1);
```

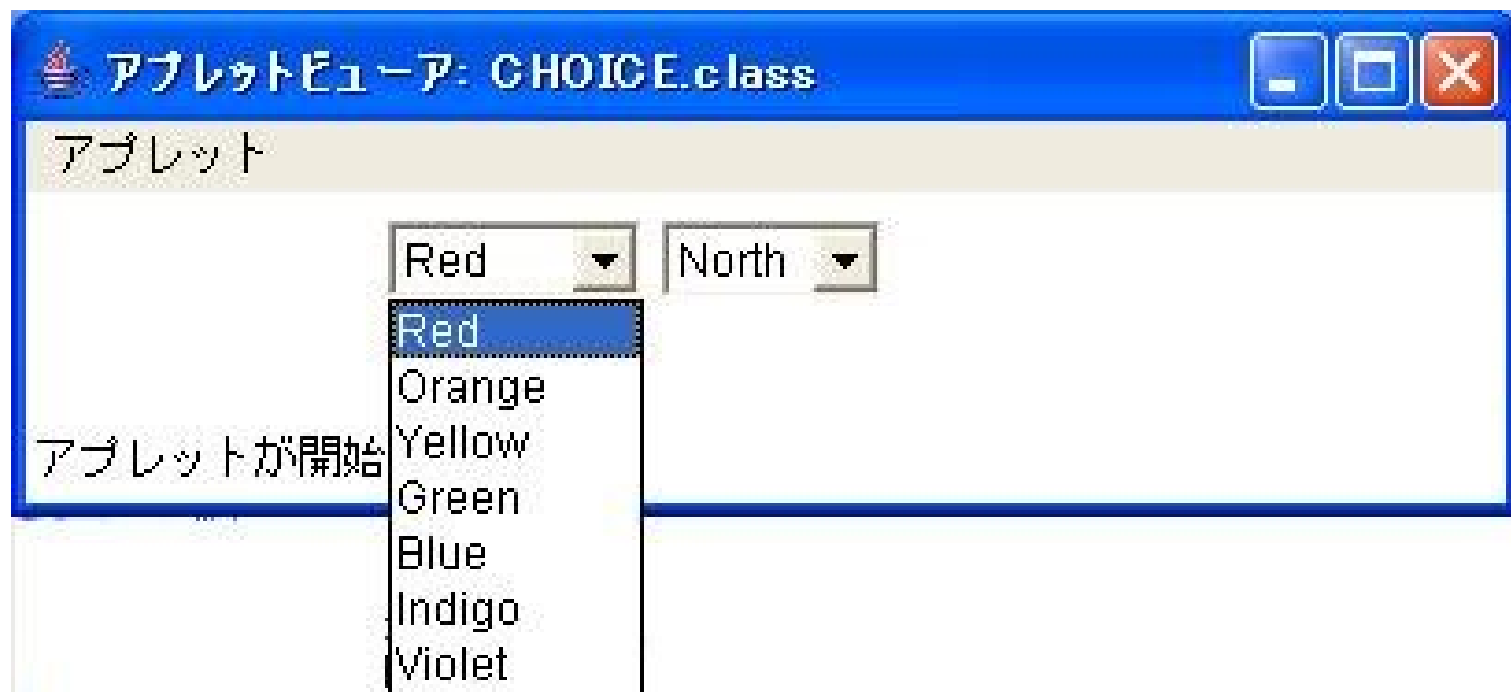


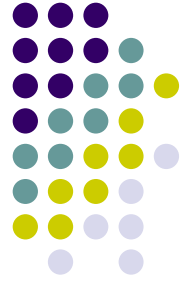
使用例(2/2)

```
Choice c2 = new  
Choice();  
c2.addItem("North");  
c2.addItem("South");  
c2.addItem("East");  
c2.addItem("West");  
c2.addItemListener(this);  
add(c2);  
label = new Label("  
");  
add(label);  
}
```

```
public void  
itemStateChanged(ItemEv  
ent ie){  
    Choice c =  
(Choice)ie.getItemSelecta  
ble();  
  
    label.setText(c.getSele  
ctedItem());  
}  
}
```

実行結果





練習問題

- チェックボックス、チョイスを用いて、描画する図形の種類、色、位置を指定し、ボタンを押すことにより、描画を行うアプレットを作成しなさい
 - チェックボックス
 - red, green, blueで色指定(組み合わせあり)
 - チェックボックスグループ
 - Oval, Rect, Polygonの3つの中から種類を指定
 - チョイス
 - 項目により、表示位置を指定(上下左右など)