

イベントリスナ (アクションの検知)

Exec300.java

```
1: // add Message Label   メッセージ出力
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.FlowLayout;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.BoxLayout;
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JPanel;
14: public class Exec300 extends JFrame implements ActionListener{
15:     JFrame execFrame;
16:     JPanel copyPanel;
17:     public Exec300(){
18:         execFrame = new JFrame("メッセージ出力");
19:         execFrame.setSize(600,400);
20:         execFrame.setVisible(true);
21:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22:
23:         JPanel buttonPanel = new JPanel();
24:         JButton messageButton = new JButton("メッセージ");
25:         buttonPanel.add(new JLabel("ボタンを"));
26:         buttonPanel.add(new JLabel("クリック"));
27:         buttonPanel.add(messageButton);
28:
29:         copyPanel = new JPanel();
30:         copyPanel.setLayout( new BoxLayout(copyPanel, BoxLayout.Y_AXIS));
31:
32:         Container contentPane = getContentPane();
33:         contentPane.add(buttonPanel, BorderLayout.NORTH);
34:         contentPane.add(copyPanel, BorderLayout.CENTER);
35:
36:         execFrame.setLayout( new FlowLayout()); //コメントで左端
37:         execFrame.add(contentPane);
38:         execFrame.setVisible(true);
39:
40:         messageButton.addActionListener(this); //koko
41:     }
42:     public void actionPerformed(ActionEvent e){ //koko
43:         for(int i=1;i<=5;i++){
44:             copyPanel.add(new JLabel(i+" Hello Swing World !!")); //koko
45:         }
46:         copyPanel.add(new JLabel("")); //koko
47:         execFrame.setVisible(true); //koko
48:         System.out.println("Button Clicked!"); //koko
49:     }
50:     public static void main(String[] args){
51:         new Exec300();
52:     }
53: }
54:
55:
56:
```

Exec301.java

```
1: // add Message Label 円ドル変換
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.FlowLayout;
6: import java.awt.Font;
7: import java.awt.event.ActionEvent;
8: import java.awt.event.ActionListener;
9:
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JPanel;
14: import javax.swing.JTextField;
15: public class Exec301 extends JFrame implements ActionListener {
16:     JFrame execFrame;
17:     JPanel copyPanel;
18:     JTextField rateTextField; //koko
19:     JTextField yenTextField; //koko
20:     public Exec301() {
21:         execFrame = new JFrame("円ドル変換");
22:         execFrame.setSize(600, 300);
23:         execFrame.setVisible(true);
24:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25:
26:         JPanel buttonPanel = new JPanel();
27:         JButton calcButton = new JButton("計算");
28:         rateTextField = new JTextField(10);
29:         yenTextField = new JTextField(10);
30:
31:         buttonPanel.add(new JLabel("1 ドルが"));
32:         buttonPanel.add(rateTextField);
33:         buttonPanel.add(new JLabel("円のとき"));
34:         buttonPanel.add(yenTextField);
35:         buttonPanel.add(new JLabel("円は何ドルか"));
36:         buttonPanel.add(calcButton);
37:
38:         copyPanel = new JPanel();
39:         // copyPanel.setLayout( new BorderLayout(copyPanel, BorderLayout.Y_AXIS));
40:         copyPanel.setLayout( new FlowLayout());
41:
42:         Container contentPane = getContentPane();
43:         contentPane.add(buttonPanel, BorderLayout.NORTH);
44:         contentPane.add(copyPanel, BorderLayout.CENTER);
45:
46:         execFrame.setLayout( new FlowLayout()); //コメントで左端
47:         execFrame.add(contentPane);
48:         execFrame.setVisible(true);
49:
50:         calcButton.addActionListener(this); //koko
51:     }
52:     public void actionPerformed(ActionEvent e) { //koko
53:         double rate= Double.parseDouble(rateTextField.getText());
54:         int yen= Integer.parseInt(yenTextField.getText());
55:         double dollar = yen/rate;
56:         double round=((double)Math.round(dollar * 100))/100; //小数点3位で四捨五入
57:         JLabel result= new JLabel(String.valueOf(round)+"ドルです");
58:         result.setFont(new Font("MS ゴシック", Font.BOLD, 20));
59:         copyPanel.add(result); //koko
60:         execFrame.setVisible(true); //koko
61:         System.out.println("Button Clicked!"); //koko
62:     }
63:     public static void main(String[] args) {
64:         new Exec301();
65:     }
66: }
67:
68:
69:
```

Exec310.java

```
1: //add Password field to Label パスワード表示
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.FlowLayout;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.BoxLayout;
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JPanel;
14: import javax.swing.JPasswordField;
15: public class Exec310 extends JFrame implements ActionListener {
16:     JPasswordField myPassword; //koko
17:     JFrame execFrame;
18:     JPanel copyPanel;
19:     public Exec310() {
20:         execFrame = new JFrame("パスワード表示");
21:         execFrame.setSize(400, 300);
22:         execFrame.setVisible(true);
23:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24:
25:         JPanel buttonPanel = new JPanel();
26:         JButton copyButton = new JButton("表示"); //koko
27:         myPassword = new JPasswordField(10); //koko
28:         buttonPanel.add(new JLabel("Password")); //koko
29:         buttonPanel.add(myPassword); //koko
30:         buttonPanel.add(copyButton); //koko
31:
32:         copyPanel = new JPanel();
33:         copyPanel.setLayout(new BoxLayout(copyPanel, BoxLayout.Y_AXIS));
34:
35:         Container contentPane = getContentPane();
36:         contentPane.add(buttonPanel, BorderLayout.NORTH);
37:         contentPane.add(copyPanel, BorderLayout.CENTER);
38:
39:         execFrame.setLayout(new FlowLayout());
40:         execFrame.add(contentPane);
41:         execFrame.setVisible(true);
42:
43:         copyButton.addActionListener(this); //koko
44:     }
45:     public void actionPerformed(ActionEvent e) { //koko
46:         char[] password = myPassword.getPassword(); //koko
47:         String passwordstr = new String(password); //koko
48:         copyPanel.add(new JLabel(passwordstr)); //koko
49:         execFrame.setVisible(true); //koko
50:         System.out.println(passwordstr); //koko
51:     }
52:
53:     public static void main(String[] args) {
54:         new Exec310();
55:     }
56: }
57:
58:
```

Exec311.java

```
1: //add Password Field to Label and Clear パスワード表示とクリア
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.FlowLayout;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.BoxLayout;
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JPanel;
14: import javax.swing.JPasswordField;
15: public class Exec311 extends JFrame implements ActionListener{
16:     JPasswordField myPassword; //koko
17:     JFrame execFrame;
18:     JPanel copyPanel;
19:     JButton copyButton; //kokokoko
20:     JButton clearButton; //kokokoko
21:
22:     public Exec311(){
23:         execFrame = new JFrame("パスワード表示");
24:         execFrame.setSize(400,300);
25:         execFrame.setVisible(true);
26:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27:
28:         JPanel buttonPanel = new JPanel();
29:         copyButton = new JButton("表示"); //kokokoko
30:         clearButton = new JButton("クリア"); //kokokoko
31:         myPassword = new JPasswordField(10); //koko
32:         buttonPanel.add(new JLabel("Password")); //koko
33:         buttonPanel.add(myPassword); //koko
34:         buttonPanel.add(copyButton); //koko
35:         buttonPanel.add(clearButton); //kokokoko
36:
37:         copyPanel = new JPanel();
38:         copyPanel.setLayout( new BoxLayout(copyPanel, BoxLayout.Y_AXIS));
39:
40:         Container contentPane = getContentPane();
41:         contentPane.add(buttonPanel, BorderLayout.NORTH);
42:         contentPane.add(copyPanel, BorderLayout.CENTER);
43:
44:         execFrame.setLayout( new FlowLayout());
45:         execFrame.add(contentPane);
46:         execFrame.setVisible(true);
47:
48:         copyButton.addActionListener(this); //koko
49:         clearButton.addActionListener(this); //kokokoko
50:     }
51:     public void actionPerformed(ActionEvent e){ //koko
52:         char[] password = myPassword.getPassword(); //koko
53:         String passwordstr = new String(password); //koko
54:         // ユーザの操作対象を判断
55:         if(e.getSource() == copyButton){ //kokokoko
56:             copyPanel.add(new JLabel(passwordstr)); //koko
57:         } //kokokoko
58:         if(e.getSource() == clearButton){ //kokokoko
59:             myPassword.setText(null); //kokokoko
60:         }
61:         execFrame.setVisible(true); //koko
62:         System.out.println(passwordstr); //koko
63:     }
64:
65:     public static void main(String[] args){
66:         new Exec311();
67:     }
68: }
```

Exec312.java

```

1: //パスワード表示とスクロール
2: //add Password Field to TextArea+ScrollPane and All Clear
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.FlowLayout;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.BoxLayout;
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JPanel;
14: import javax.swing.JPasswordField;
15: import javax.swing.JScrollPane;
16: import javax.swing.JTextArea;
17: public class Exec312 extends JFrame implements ActionListener{
18:     JPasswordField myPassword; //koko
19:     JFrame execFrame;
20:     JPanel copyPanel;
21:     JButton copyButton; //kokokoko
22:     JButton clearButton; //kokokoko
23:     JTextArea copyTextArea; //kokokokokokokokokoko
24:
25:     public Exec312(){
26:         execFrame = new JFrame("パスワード表示");
27:         execFrame.setSize(400, 300);
28:         execFrame.setVisible(true);
29:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
30:
31:         JPanel buttonPanel = new JPanel();
32:         copyButton = new JButton("表示"); //kokokoko
33:         clearButton = new JButton("クリア"); //kokokoko
34:         myPassword = new JPasswordField(10); //koko
35:         buttonPanel.add(new JLabel("Password")); //koko
36:         buttonPanel.add(myPassword); //koko
37:         buttonPanel.add(copyButton); //koko
38:         buttonPanel.add(clearButton); //kokokoko
39:         //テキストパネル
40:         copyPanel = new JPanel();
41:         copyPanel.setLayout(new BoxLayout(copyPanel, BoxLayout.Y_AXIS));
42:         copyTextArea = new JTextArea(10, 5); //kokokokokokokokokoko
43:         copyPanel.add(copyTextArea); //kokokokokokokokokoko
44:         //スクロール パネル
45:         copyPanel = new JPanel();
46:         copyPanel.setLayout(new BoxLayout(copyPanel, BoxLayout.Y_AXIS));
47:         copyTextArea = new JTextArea(10, 5);
48:         JScrollPane scrollPanel = new JScrollPane(copyTextArea);
49:         copyPanel.add(scrollPanel);
50:
51:         Container contentPane = getContentPane();
52:         contentPane.add(buttonPanel, BorderLayout.NORTH);
53:         contentPane.add(copyPanel, BorderLayout.CENTER);
54:
55:         execFrame.setLayout(new FlowLayout());
56:         execFrame.add(contentPane);
57:         execFrame.setVisible(true);
58:
59:         copyButton.addActionListener(this); //koko
60:         clearButton.addActionListener(this); //kokokoko
61:     }
62:     public void actionPerformed(ActionEvent e){ //koko
63:         char[] password = myPassword.getPassword(); //koko
64:         String passwordstr = new String(password); //koko
65:         // ユーザの操作対象を判断
66:         if(e.getSource() == copyButton){ //kokokoko
67:             // テキストエリアへ文字列を追加
68:             copyTextArea.append(passwordstr + "\n"); //kokokokokokokokokoko
69:         } //kokokoko
70:         if(e.getSource() == clearButton){ //kokokoko
71:             myPassword.setText(null); //kokokoko
72:             copyTextArea.setText(null); //kokokoko
73:         }
74:         execFrame.setVisible(true); //koko
75:         System.out.println(passwordstr); //koko
76:     }
77:
78:     public static void main(String[] args){
79:         new Exec312();
80:     }
81: }
82:
83:

```

Exec320.java

```
1: //CheckBox + Dialogue チェックボックスとダイアログ
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.event.ActionEvent;
6: import java.awt.event.ActionListener;
7:
8: import javax.swing.BoxLayout;
9: import javax.swing.JButton;
10: import javax.swing.JCheckBox;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JOptionPane;
14: import javax.swing.JPanel;
15:
16: public class Exec320 extends JFrame implements ActionListener{
17:     JCheckBox[] check; //koko
18:     JFrame execFrame;
19:     JPanel confirmPanel; //koko
20:     public Exec320(){
21:         execFrame = new JFrame("チェックボックス ダイアログ");
22:         // execFrame.setLayout( new FlowLayout());
23:         // execFrame.add(new JCheckBox("I am CheckBox"));
24:         execFrame.setSize(400,300);
25:         execFrame.setVisible(true);
26:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27:         //kokokara
28:         check = new JCheckBox[3];
29:         check[0] = new JCheckBox("Travel");
30:         check[1] = new JCheckBox("Movie");
31:         check[2] = new JCheckBox("Sports");
32:
33:         JPanel checkPanel = new JPanel();
34:         checkPanel.add(check[0]);
35:         checkPanel.add(check[1]);
36:         checkPanel.add(check[2]);
37:
38:         confirmPanel = new JPanel(); //koko
39:         // confirmPanel.setLayout( new GridLayout(3,1));
40:         confirmPanel.setLayout( new BoxLayout(confirmPanel, BoxLayout.Y_AXIS));
41:
42:         JButton button = new JButton("Check");
43:         button.addActionListener(this);
44:
45:         JPanel buttonPanel = new JPanel();
46:         buttonPanel.add(button);
47:
48:         Container contentPane = getContentPane();
49:         contentPane.add(checkPanel, BorderLayout.NORTH); //koko
50:         contentPane.add(confirmPanel, BorderLayout.CENTER); //koko
51:         contentPane.add(buttonPanel, BorderLayout.SOUTH);
52:
53:         execFrame.add(contentPane);
54:         execFrame.setVisible(true);
55:     }
56:     public void actionPerformed(ActionEvent e){
57:
58:         StringBuilder sb = new StringBuilder();
59:
60:         for (int i = 0 ; i < check.length; i++){
61:             sb.append(check[i].getText());
62:
63:             if (check[i].isSelected()){
64:                 sb.append("[Select]");
65:             }else{
66:                 sb.append("[Not Select]");
67:             }
68:             sb.append(" ");
69:         }
70:
71:         JOptionPane.showMessageDialog(this, new String(sb));
72:         confirmPanel.add(new JLabel(new String(sb))); //koko
73:         execFrame.setVisible(true);
74:     }
75:     public static void main(String[] args){
76:         new Exec320();
77:     }
78: }
```

Exec321.java

```
1: //RadioButton + Dialogue ラジオボタンとダイアログ
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.event.ActionEvent;
6: import java.awt.event.ActionListener;
7:
8: import javax.swing.BoxLayout;
9: import javax.swing.ButtonGroup;
10: import javax.swing.JButton;
11: import javax.swing.JFrame;
12: import javax.swing.JLabel;
13: import javax.swing.JOptionPane;
14: import javax.swing.JPanel;
15: import javax.swing.JRadioButton;
16:
17: public class Exec321 extends JFrame implements ActionListener{ //koko
18:     JRadioButton[] radio; //koko
19:     JFrame execFrame;
20:     JPanel confirmPanel; //koko
21:     public Exec321(){
22:         execFrame = new JFrame("ラジオボタン ダイアログ");
23:         // execFrame.setLayout( new FlowLayout( ));
24:         // execFrame.add(new JRadioButton("I am RadioButton"));
25:         execFrame.setSize(400, 300);
26:         execFrame.setVisible(true);
27:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28:         //kokokara
29:         radio = new JRadioButton[3];
30:
31:         radio[0] = new JRadioButton("Peach");
32:         radio[1] = new JRadioButton("Orange");
33:         radio[2] = new JRadioButton("Banana");
34:
35:         ButtonGroup bgroup = new ButtonGroup();
36:         bgroup.add(radio[0]);
37:         bgroup.add(radio[1]);
38:         bgroup.add(radio[2]);
39:
40:         JPanel radioPanel = new JPanel();
41:         radioPanel.add(radio[0]);
42:         radioPanel.add(radio[1]);
43:         radioPanel.add(radio[2]);
44:
45:         confirmPanel = new JPanel(); //koko
46:         // confirmPanel.setLayout( new GridLayout(3, 1));
47:         confirmPanel.setLayout( new BoxLayout(confirmPanel, BoxLayout.Y_AXIS));
48:
49:         JButton button = new JButton("Check");
50:         button.addActionListener(this);
51:
52:         JPanel buttonPanel = new JPanel();
53:         buttonPanel.add(button);
54:
55:         Container contentPane = getContentPane();
56:         contentPane.add(radioPanel, BorderLayout.NORTH); //koko
57:         contentPane.add(confirmPanel, BorderLayout.CENTER); //koko
58:         contentPane.add(buttonPanel, BorderLayout.SOUTH);
59:
60:         execFrame.add(contentPane);
61:         execFrame.setVisible(true);
62:     }
63:     public void actionPerformed(ActionEvent e){
64:
65:         StringBuilder sb = new StringBuilder();
66:
67:         for (int i = 0 ; i < radio.length; i++){
68:             sb.append(radio[i].getText());
69:
70:             if (radio[i].isSelected()){
71:                 sb.append("[Select]");
72:             }else{
73:                 sb.append("[Not Select]");
74:             }
75:             sb.append(" ");
76:         }
77:         JOptionPane.showMessageDialog(this, new String(sb));
78:
79:         confirmPanel.add(new JLabel(new String(sb))); //koko
80:         execFrame.setVisible(true);
81:     }
82:     public static void main(String[] args){
83:         new Exec321();
84:     }
85: }
86:
```

Exec330.java

```
1: //ComboBox View コンボボックスの項目取得
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.Dimension;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.DefaultComboBoxModel;
10: import javax.swing.JButton;
11: import javax.swing.JComboBox;
12: import javax.swing.JFrame;
13: import javax.swing.JLabel;
14: import javax.swing.JPanel;
15: import javax.swing.JScrollPane;
16: import javax.swing.JTextArea;
17: import javax.swing.JTextField;
18:
19: public class Exec330 extends JFrame implements ActionListener {
20:     JFrame execFrame;
21:     DefaultComboBoxModel model;
22:     JTextField datatext;
23:     JTextField indextext;
24:     JButton viewButton;
25:     JComboBox combo;
26:     JTextArea copyTextArea;
27:     public Exec330() {
28:         execFrame = new JFrame("コンボボックス項目取得");
29:         execFrame.setSize(400, 300);
30:         execFrame.setVisible(true);
31:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
32:         //kokokara
33:         String[] combodata = {"Red", "Yellow", "Green"};
34:         model = new DefaultComboBoxModel(combodata);
35:
36:         combo = new JComboBox(model);
37:         combo.setPreferredSize(new Dimension(100, 30));
38:
39:         viewButton = new JButton("取得");
40:         viewButton.addActionListener(this); //ActionListener
41:
42:         //コンボボックスパネル
43:         JPanel comboPanel = new JPanel();
44:         comboPanel.add(combo);
45:         comboPanel.add(viewButton);
46:
47:         indextext = new JTextField(2);
48:         datatext = new JTextField(6);
49:
50:         //コントロールパネル
51:         JPanel controlPanel = new JPanel();
52:         controlPanel.add(new JLabel("index:"));
53:         controlPanel.add(indextext);
54:         controlPanel.add(new JLabel("data:"));
55:         controlPanel.add(datatext);
56:
57:         //スクロール パネル
58:         JPanel copyPanel = new JPanel();
59:         // copyPanel.setLayout(new BorderLayout(copyPanel, BorderLayout.Y_AXIS));
60:         copyTextArea = new JTextArea();
61:         JScrollPane scrollPanel = new JScrollPane(copyTextArea);
62:         // JScrollPane scrollPanel = new JScrollPane(copyTextArea,
63:         //     JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
64:         //     JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
65:
66:         copyPanel.add(scrollPanel);
67:
68:         Container contentPane = getContentPane();
69:         getContentPane().add(comboPanel, BorderLayout.NORTH);
70:         getContentPane().add(scrollPanel, BorderLayout.CENTER);
71:         getContentPane().add(controlPanel, BorderLayout.SOUTH);
72:
73:         execFrame.add(contentPane);
74:         execFrame.setVisible(true);
75:     }
76:     public void actionPerformed(ActionEvent e) {
77:         // ユーザの操作対象を判断
78:         if (e.getSource() == viewButton) {
79:             // テキストエリアへ文字列を追加
80:             copyTextArea.append("Index:" + combo.getSelectedIndex() + "\n");
81:             copyTextArea.append("Item : " + (String) combo.getSelectedItem() + "\n\n");
82:         }
83:         execFrame.setVisible(true);
84:         System.out.println(combo.getSelectedIndex());
85:         System.out.println((String) combo.getSelectedItem());
86:     }
87: }
88: public static void main(String[] args) {
89:     new Exec330();
90: }
91: }
```

Exec331.java

```
1: //ComboBox View+Insert コンボボックスの項目追加
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.Dimension;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.DefaultComboBoxModel;
10: import javax.swing.JButton;
11: import javax.swing.JComboBox;
12: import javax.swing.JFrame;
13: import javax.swing.JLabel;
14: import javax.swing.JPanel;
15: import javax.swing.JScrollPane;
16: import javax.swing.JTextArea;
17: import javax.swing.JTextField;
18:
19: public class Exec331 extends JFrame implements ActionListener {
20:     JFrame execFrame;
21:     DefaultComboBoxModel model;
22:     JTextField datatext;
23:     JTextField indextext;
24:     JButton viewButton;
25:     JButton insertButton; //koko
26:     JComboBox combo;
27:     JTextArea copyTextArea;
28:     public Exec331() {
29:         execFrame = new JFrame("コンボボックス項目追加");
30:         execFrame.setSize(400, 300);
31:         execFrame.setVisible(true);
32:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
33:         //kokokara
34:         String[] combodata = {"Red", "Yellow", "Green"};
35:         model = new DefaultComboBoxModel(combodata);
36:
37:         combo = new JComboBox(model);
38:         combo.setPreferredSize(new Dimension(100, 30));
39:
40:         viewButton = new JButton("取得");
41:         viewButton.addActionListener(this); //ActionListener
42:
43:         //コンボボックスパネル
44:         JPanel comboPanel = new JPanel();
45:         comboPanel.add(combo);
46:         comboPanel.add(viewButton);
47:
48:         indextext = new JTextField(2);
49:         datatext = new JTextField(6);
50:
51:         //コントロールボックスパネル
52:         insertButton = new JButton("追加");
53:         insertButton.addActionListener(this); //ActionListener koko
54:
55:         JPanel controlPanel = new JPanel();
56:         controlPanel.add(new JLabel("index:"));
57:         controlPanel.add(indextext);
58:         controlPanel.add(new JLabel("data:"));
59:         controlPanel.add(datatext);
60:         controlPanel.add(insertButton); //koko
61:
62:         //スクロール パネル
63:         JPanel copyPanel = new JPanel();
64:         // copyPanel.setLayout(new BorderLayout(copyPanel, BorderLayout.Y_AXIS));
65:         copyTextArea = new JTextArea();
66:         JScrollPane scrollPanel = new JScrollPane(copyTextArea);
67:         // JScrollPane scrollPanel = new JScrollPane(copyTextArea,
68:         // JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
69:         // JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
```

```

70:
71:         copyPanel.add(scrollPanel);
72:
73:         Container contentPane = getContentPane();
74:         getContentPane().add(comboPanel, BorderLayout.NORTH);
75:         getContentPane().add(scrollPanel, BorderLayout.CENTER);
76:         getContentPane().add(controlPanel, BorderLayout.SOUTH);
77:
78:         execFrame.add(contentPane);
79:         execFrame.setVisible(true);
80:     }
81:     public void actionPerformed(ActionEvent e) {
82:         // ユーザの操作対象を判断
83:         if(e.getSource() == viewButton) {
84:             // テキストエリアへ文字列を追加
85:             copyTextArea.append("Index:" + combo.getSelectedIndex() + "¥n");
86:             copyTextArea.append("Item :"+(String)combo.getSelectedItem() + "¥n¥n");
87:         }
88:         if(e.getSource() == insertButton) { //koko
89:             // コンボボックスへ項目を追加
90:             try{
91:                 int index = Integer.parseInt(indextext.getText());
92:                 String newdata = datatext.getText();
93:
94:                 model.insertElementAt(newdata, index);
95:                 indextext.setText("");
96:                 datatext.setText("");
97:             } catch (NumberFormatException ex) {
98:                 copyTextArea.append("Bad TextField Error!" + "¥n");
99:                 return;
100:             } catch (ArrayIndexOutOfBoundsException ex) {
101:                 copyTextArea.append("Bad Index Error!" + "¥n");
102:                 return;
103:             }
104:         }
105:         execFrame.setVisible(true);
106:         System.out.println(combo.getSelectedIndex());
107:         System.out.println((String)combo.getSelectedItem());
108:
109:     }
110: }
111: public static void main(String[] args) {
112:     new Exec331();
113: }
114: }

```

Exec332.java

```
1: //ComboBox View+Insert+Remove コンボボックスの項目削除
2:
3: import java.awt.BorderLayout;
4: import java.awt.Container;
5: import java.awt.Dimension;
6: import java.awt.event.ActionEvent;
7: import java.awt.event.ActionListener;
8:
9: import javax.swing.DefaultComboBoxModel;
10: import javax.swing.JButton;
11: import javax.swing.JComboBox;
12: import javax.swing.JFrame;
13: import javax.swing.JLabel;
14: import javax.swing.JPanel;
15: import javax.swing.JScrollPane;
16: import javax.swing.JTextArea;
17: import javax.swing.JTextField;
18:
19: public class Exec332 extends JFrame implements ActionListener {
20:     JFrame execFrame;
21:     DefaultComboBoxModel model;
22:     JTextField datatext;
23:     JTextField indextext;
24:     JButton viewButton;
25:     JButton insertButton; //koko
26:     JButton removeButton; //koko
27:     JButton clearButton; //koko
28:     JComboBox combo;
29:     JTextArea copyTextArea;
30:     public Exec332() {
31:         execFrame = new JFrame("コンボボックス項目削除");
32:         execFrame.setSize(400, 300);
33:         execFrame.setVisible(true);
34:         execFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35:         //kokokara
36:         String[] combodata = {"Red", "Yellow", "Green"};
37:         model = new DefaultComboBoxModel(combodata);
38:
39:         combo = new JComboBox(model);
40:         combo.setPreferredSize(new Dimension(100, 30));
41:
42:         viewButton = new JButton("取得");
43:         viewButton.addActionListener(this); //ActionListener
44:
45:         clearButton = new JButton("クリア"); //koko
46:         clearButton.addActionListener(this); //ActionListener koko
47:
48:         //コンボボックスパネル
49:         JPanel comboPanel = new JPanel();
50:         // comboPanel.setLayout(new BoxLayout(comboPanel, BoxLayout.Y_AXIS));
51:         // comboPanel.add(combo);
52:         // comboPanel.add(viewButton);
53:         // comboPanel.add(clearButton); //koko
54:
55:         indextext = new JTextField(2);
56:         datatext = new JTextField(6);
57:
58:         //コントロールボックスパネル
59:         insertButton = new JButton("追加");
60:         insertButton.addActionListener(this); //ActionListener koko
61:
62:         removeButton = new JButton("削除");
63:         removeButton.addActionListener(this); //ActionListener koko
64:
65:         JPanel controlPanel = new JPanel();
66:         controlPanel.add(new JLabel("index:"));
67:         controlPanel.add(indextext);
68:         controlPanel.add(new JLabel("data:"));
69:     }
}
```

```

70:         controlPanel.add(datatext);
71:         controlPanel.add(insertButton);           //koko
72:         controlPanel.add(removeButton);          //koko
73:
74:         //スクロール パネル
75:         JPanel copyPanel = new JPanel();
76:         //         copyPanel.setLayout( new BorderLayout( copyPanel, BorderLayout.Y_AXIS));
77:         copyTextArea = new JTextArea(10, 10);
78:         JScrollPane scrollIPanel = new JScrollPane(copyTextArea);
79:         //         JScrollPane scrollIPanel = new JScrollPane(copyTextArea,
80:         //             JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
81:         //             JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
82:
83:         copyPanel.add(scrollIPanel);
84:
85:         Container contentPane = getContentPane();
86:         getContentPane().add(comboPanel, BorderLayout.NORTH);
87:         getContentPane().add(scrollIPanel, BorderLayout.CENTER);
88:         getContentPane().add(controlPanel, BorderLayout.SOUTH);
89:
90:         execFrame.add(contentPane);
91:         execFrame.setVisible(true);
92:     }
93:     public void actionPerformed(ActionEvent e) {
94:         // ユーザの操作対象を判断
95:         if(e.getSource() == viewButton) {
96:             // テキストエリアへ文字列を追加
97:             copyTextArea.append("Index:"+combo.getSelectedIndex() + "\n");
98:             copyTextArea.append("Item :"+(String)combo.getSelectedItem() + "\n\n");
99:         }
100:         if(e.getSource() == clearButton) {        //koko
101:             // テキストエリアをクリア
102:             copyTextArea.setText(null);
103:             // データエリアをクリア
104:             indextext.setText(null);
105:             datatext.setText(null);
106:         }
107:         if(e.getSource() == insertButton) {        //koko
108:             // コンボボックスへ項目を追加
109:             try{
110:                 int index = Integer.parseInt(indextext.getText());
111:                 String newdata = datatext.getText();
112:
113:                 model.insertElementAt(newdata, index);
114:                 indextext.setText("");
115:                 datatext.setText("");
116:             } catch (NumberFormatException ex) {
117:                 copyTextArea.append("Bad TextField Error!"+"\n");
118:                 return;
119:             } catch (ArrayIndexOutOfBoundsException ex) {
120:                 copyTextArea.append("Bad Index Error!"+"\n");
121:                 return;
122:             }
123:         }
124:
125:         if(e.getSource() == removeButton) {        //koko
126:             // コンボボックスの項目を削除
127:             try{
128:                 int index = Integer.parseInt(indextext.getText());
129:                 String newdata = datatext.getText();
130:
131:                 model.removeElementAt(index);        //koko
132:                 indextext.setText("");
133:                 datatext.setText("");
134:             } catch (NumberFormatException ex) {
135:                 copyTextArea.append("Bad TextField Error!"+"\n");
136:                 return;
137:             } catch (ArrayIndexOutOfBoundsException ex) {
138:                 copyTextArea.append("Bad Index Error!"+"\n");
139:                 return;
140:             }
141:
142:         }
143:         execFrame.setVisible(true);
144:         System.out.println(combo.getSelectedIndex());
145:         System.out.println((String)combo.getSelectedItem());
146:     }
147: }
148: public static void main(String[] args){
149:     new Exec332();
150: }
151: }

```