

```

Loading vc-cvs...
1 package caltool.schedule_ui;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.*;
6
7 public class RecurringCheckBoxListener implements ActionListener {
8
9     public RecurringCheckBoxListener(RecurringInfoSubdialog parentView,
10         ScheduleAppointmentDialog grandparentView) {
11         this.parentView = parentView;
12         this.grandparentView = grandparentView;
13     }
14
15     public void actionPerformed(ActionEvent e) {
16
17         if (parentView.recurringCheckBox.isSelected()) {
18             if ((grandparentView.getClass() ==
19                 ScheduleAppointmentDialog.class) ||
20                 (grandparentView.getClass() ==
21                 ScheduleTaskDialog.class)) {
22                 grandparentView.startDateLabel.setText("Start Date: ");
23             }
24             else /* grandparent is meeting dialog */ {
25                 grandparentView.startDateLabel.setText(
26                     "Earliest Start Date: ");
27                 ((ScheduleMeetingDialog)grandparentView).latestStartDateLabel.
28                     setText("Latest Start Date: ");
29             }
30             setEnabled(true);
31         }
32         else {
33             if (grandparentView.getClass() ==
34                 ScheduleAppointmentDialog.class) {
35                 grandparentView.startDateLabel.setText("Date: ");
36             }
37             else if (grandparentView.getClass() == ScheduleTaskDialog.class) {
38                 grandparentView.startDateLabel.setText("Due Date: ");
39             }
40             else /* grandparent is meeting dialog */ {
41                 grandparentView.startDateLabel.setText("Earliest Date: ");
42                 ((ScheduleMeetingDialog)grandparentView).latestStartDateLabel.
43                     setText("Latest Date: ");
44             }
45             setEnabled(false);
46             clearAll();
47         }
48     }
49
50     protected void setEnabled(boolean isEnabled) {
51         grandparentView.endDateLabel.setEnabled(isEnabled);
52         grandparentView.endDateTextField.setEnabled(isEnabled);
53         if (grandparentView.getClass() == ScheduleMeetingDialog.class) {
54             ((ScheduleMeetingDialog)grandparentView).latestEndDateLabel.
55                 setEnabled(isEnabled);
56             ((ScheduleMeetingDialog)grandparentView).latestEndDateTextField.
57                 setEnabled(isEnabled);
58         }
59
60         parentView.intervalComboBox.setEnabled(isEnabled);
61         parentView.intervalLabel.setEnabled(isEnabled);
62         parentView.sunCheckBox.setEnabled(isEnabled);
63         parentView.sunLabel.setEnabled(isEnabled);
64         parentView.monCheckBox.setEnabled(isEnabled);
65         parentView.monLabel.setEnabled(isEnabled);
66         parentView.tueCheckBox.setEnabled(isEnabled);
67         parentView.tueLabel.setEnabled(isEnabled);
68         parentView.wedCheckBox.setEnabled(isEnabled);
69         parentView.wedLabel.setEnabled(isEnabled);
70         parentView.thuCheckBox.setEnabled(isEnabled);
71         parentView.thuLabel.setEnabled(isEnabled);
72         parentView.friCheckBox.setEnabled(isEnabled);
73         parentView.friLabel.setEnabled(isEnabled);
74         parentView.satCheckBox.setEnabled(isEnabled);
75         parentView.satLabel.setEnabled(isEnabled);
76     }
77
78     protected void clearAll() {
79         parentView.sunCheckBox.setSelected(false);
80         parentView.monCheckBox.setSelected(false);
81         parentView.tueCheckBox.setSelected(false);
82         parentView.wedCheckBox.setSelected(false);
83         parentView.thuCheckBox.setSelected(false);
84         parentView.friCheckBox.setSelected(false);
85         parentView.satCheckBox.setSelected(false);
86     }
87
88     /** The recurring info parent view */
89     protected RecurringInfoSubdialog parentView;
90
91     /** The schedule appointment grandparent view */
92     protected ScheduleAppointmentDialog grandparentView;
93
94
95 }

```