

package com.mycompany.eventprog;

import static com.codename1.ui.CN.\*;

import com.codename1.ui.Display;

import com.codename1.ui.Form;

import com.codename1.ui.Dialog;

import com.codename1.ui.Label;

import com.codename1.ui.plaf.UIManager;

import com.codename1.ui.util.Resources;

import com.codename1.io.Log;

import com.codename1.ui.Toolbar;

import java.io.IOException;

import com.codename1.ui.layouts.BoxLayout;

/\*\*

\* This file was generated by <a href="https://www.codenameone.com/">Codename One</a> for the purpose

\* of building native mobile applications using Java.

\*/

public class Starter {

private Form current;

private Resources theme;

public void init(Object context) {

theme = UIManager.initFirstTheme("/theme");

// Enable Toolbar on all Forms by default

Toolbar.setGlobalToolbar(true);

// Pro only feature, uncomment if you have a pro subscription

Log.bindCrashProtection(true);

}

public void start() {

if(current != null){

current.show();

return;

}

// 1s Demo - ButtonListenerForm

// new ButtonListenerForm();

// 2nd Demo- SelfListenerForm

// new SelfListenerForm ();

// 3rd Demo - MultipleComponentListener1b

// new MultipleComponentListener1b ();

// 4th Demo - SideMenuItemCheckFormUsingCommand

// new SideMenuItemCheckFormUsingCommand ();

// 5th Demo - FormWithMultipleContainers with show

new FormWithMultipleContainers();

// 6th Demo - PointerListenerForm

// new PointerListenerForm ();

// 7th Demo - PointerListenerFormWithOverride

// new PointerListenerFormWithOverride();

}

public void stop() {

current = getCurrentForm();

if(current instanceof Dialog) {

((Dialog)current).dispose();

current = getCurrentForm();

}

}

public void destroy() {

}

}

package com.mycompany.eventprog;

import com.codename1.charts.util.ColorUtil;

import com.codename1.ui.Button;

import com.codename1.ui.CheckBox;

import com.codename1.ui.ComboBox;

import com.codename1.ui.Component;

import com.codename1.ui.Container;

//Contents of File SimpleForm.java:

import com.codename1.ui.Form;

import com.codename1.ui.Label;

import com.codename1.ui.layouts.BorderLayout;

import com.codename1.ui.layouts.BoxLayout;

import com.codename1.ui.layouts.FlowLayout;

import com.codename1.ui.layouts.GridLayout;

import com.codename1.ui.plaf.Border;

import com.codename1.ui.Button;

import com.codename1.ui.Container;

import com.codename1.ui.Command;

import com.codename1.ui.events.ActionEvent;

public class FormWithMultipleContainers extends Form {

Container centerContainer, bottomContainer;

public FormWithMultipleContainers(){

//create the center container and add it to form

/\* Code for a form with containers in different layout arrangements \*/

setLayout(new BorderLayout());

//left Container with the BoxLayout positioned on the west

Container leftContainer = new PointerClickContainer();

leftContainer.add(new Label("Click Me"));

leftContainer.getAllStyles().setBorder(Border.createLineBorder(4,

ColorUtil.BLUE));

add(BorderLayout.WEST,leftContainer);

//right Container with the GridLayout positioned on the east

Container rightContainer = new PointerReleaseContainer();

rightContainer.add(new Label("Release Here"));

rightContainer.getAllStyles().setBorder(Border.createLineBorder(4,

ColorUtil.BLUE));

add(BorderLayout.EAST,rightContainer);

//add empty container to the center

Container centerContainer = new PointerDragContainer();

//setting the back ground color of center container to light gray

centerContainer.getAllStyles().setBgTransparency(255);

centerContainer.getAllStyles().setBgColor(ColorUtil.LTGRAY);

centerContainer.add(new Label("Drag Through Me"));

//setting the border Color

centerContainer.getAllStyles().setBorder(Border.createLineBorder(4,

ColorUtil.MAGENTA));

add(BorderLayout.CENTER,centerContainer);

this.show();

}// constructor

}

package com.mycompany.eventprog;

import com.codename1.ui.Container;

public class PointerClickContainer extends Container{

@Override

public void pointerPressed(int x,int y){

System.out.println("Pointer PRESSED x and y: " + x + " " + y); }

}

package com.mycompany.eventprog;

import com.codename1.ui.Container;

public class PointerDragContainer extends Container{

@Override

public void pointerDragged(int x,int y){

System.out.println("Pointer DRAGGED x and y: " + x + " " + y); }

}

package com.mycompany.eventprog;

import com.codename1.ui.Container;

public class PointerReleaseContainer extends Container{

@Override

public void pointerReleased(int x,int y){

System.out.println("Pointer RELEASED x and y: " + x + " " + y); }

}