**Week 8 – DQ1: Introduction**

Author: Daniel Schieberle  
Last Update: August 23rd 2013  
Java Version: 1.6  
Book Edition: 9

General comments:

I opted to pre-load the images using an additional thread. The pictures are taken from <http://www.bayern.by> and give some impressions of my home country, Bavaria in Germany.

BackgroundLoader.java

Instead of implementing the Runnable interface, I opted to extend Java’s “SwingWorker” class to outsource image pre-loading into an additional thread. The SwingWorker class itself extends the Thread class and ensures that GUI operations are handled within the EDT and everything that is supposed to operate in the background is located in the “doInBackground” method.  
As a result, the GUI stays responsive (updating a progressbar in this case) while the pictures are being loaded. To show-case the loading process, I added a “Thread.sleep(100)” to slow loading.

JPicturePane.java

A class extending JPanel that has 2 visual components. I used the CardLayout Manager this time to allow panels being brought to the front (like in a card deck).   
Primarily there are two panels labeled “PREVIEW” and “PCITURES” that are located on top of each other. When the application starts, the PREVIEW panel is displayed in the front (holding a progress bar). As soon as the images are loaded, the “PICTURES” panel is brought to the front.  
The same principle applies to the Images being added by the “BackgroundLoader” class. For each image, a card-layouted panel is added to the “PICTURES” panel that at the end are all stacked on top of each other. A public method “showImage” accepting an index as parameter will handle the panel flipping and bring the associated picture to the front.  
Additionally the JPicturePane holds a custom ActionListener implementation that allows the “SchieberleWeek8DQ1” class instance to be notified, once all pictures are loaded.  
In the current approach, the event is used to enable the JSlider component taken from the lecture’s example.