

Übungsblatt 2: MVC & Observer

Aufgabe 1: Observer

20 Punkte

Nachdem auf dem letzten Übungsblatt die GUI entworfen wurde, soll jetzt eine erste Funktionalität implementiert werden. Zur Implementierung der Funktionalität soll eine **Model-View-Controller** Architektur genutzt werden, wobei die View über ein **Observer-Pattern** aktualisiert wird.

Folgende Funktionen soll das Programm bieten:

- **Datei laden:** Über den entsprechenden Eintrag im Menü soll eine Bild-Datei geöffnet und anschließend in der GUI angezeigt werden.
- Das neu geladene Bild soll sowohl im linken als auch im rechten „Slot“ angezeigt werden. Achten Sie auf die korrekte **Skalierung** des Bildes bzw. des Frames.
- Sobald eine (neue) Datei geöffnet wurde, soll der **Name der Datei in der Titelleiste** des Fensters entsprechend aktualisiert werden.
- Der Menüpunkt „**Exit**“ im Menü soll das Programm beenden.
- Der Menüpunkt „**About Medientechnik**“ liefert eine kurze Info, z.B. zum Programm-Autor.
- Die Checkboxen im Menüpunkt „**Filters**“ sind einzeln (de-)aktivierbar. Da noch keine Auswirkungen (sprich: Bildfilter) implementiert werden, reicht es vorerst, wenn im Model des Programms entsprechende Variablen gesetzt werden, die verdeutlichen, dass die Filter aktiv sind.
- Der Button „**Alle Effekte**“ ermöglicht die (De-)Aktivierung aller Effekte.
Besonderheit: Sind bereits Filter aktiv, bleiben diese natürlich aktiviert, wird der „Alle Effekte“ Button aber deaktiviert, sollen die *zuvor aktiven Filter weiterhin aktiv* bleiben!
- Der Button „**Random**“ aktiviert *einen* zufälligen Effekt, alle anderen werden deaktiviert.
- Im „Filters“ **Textfeld** werden immer die Namen aller aktiven Filter angezeigt.

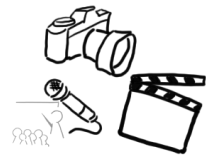
Im Model des Programms sollen dabei u.a. folgende Informationen gespeichert werden:

- das angezeigte Bild (inkl. des Namens)
- welche Effekte aktiviert sind (z.B. repräsentiert durch mehrere Bool-Variablen)

Nutzen Sie außerdem Javadoc (<https://docs.oracle.com/javase/1.5.0/docs/guide/javadoc/index.html>, <http://www.java-doc.de/>), um Ihren Code sinnvoll zu kommentieren und eine eigene API generieren zu können!



Lesen Sie sich Kapitel 6 (S.61 -76) im Buch „[Weniger schlecht programmieren](#)“ von K. Passig & J. Jander (2013) durch, um die Best Practices zum Thema Kommentare kennenzulernen.



Geben Sie die erstellten Dateien gepackt (zip, kein .rar) mit dem Namen „aufgabe1.zip“ ab. Halten Sie sich dabei unbedingt an das **vorgegebene Template auf der Website (bzgl. Dateinamen, Ordnerstruktur, ...) – es werden **ausschließlich lauffähige Lösungen im korrekten Format bewertet!****

Abgabe: Packen Sie alle Dateien dieses Übungsblatts in eine Datei („blatt2.zip“) und geben Sie diese bis spätestens **14.05.2015, 18:00 Uhr (MESZ)** über UniWorX ab! Eine spätere Abgabe oder eine Abgabe per E-Mail ist nicht möglich!

Da dieses Übungsblatt auf den Ergebnissen des ersten Blatts aufbaut, wird eine entsprechende Musterlösung von Blatt 1 nach dessen Abgabefrist (07.05.) auf der Homepage zur Verfügung gestellt!

Bei Problemen oder Fragen können *die-informatiker.net*, die Tutoren oder die Übungsleitung kontaktiert werden.