

Medientechnik - Programmierung 1

Aufgaben

Aufgabe 1: Design Patterns und Anti-Patterns (6 Punkte)

In Softwareprojekten finden sich viele Aufgaben und Situationen, die untereinander sehr ähnlich sind. Um sich nicht jedes Mal wieder Gedanken machen zu müssen, wie man am Besten vorgehen soll wurden eine Reihe von Design Patterns (deutsch: "Entwurfsmuster") entwickelt, abstrakte Beschreibungen wie ein Problem in einer objektorientierten Sprache gelöst werden kann.

Sie haben in den Übungen bereits die beiden Muster Model-View-Controller und Observer kennengelernt.

- Suchen Sie im Internet nach Informationen zu Design Patterns. Wählen Sie drei bekannte Muster aus und beschreiben Sie jedes mit mindestens drei Sätzen in Ihren eigenen (!) Worten.
- "Antipatterns" beschreiben satirisch Pseudo-Entwurfsmuster, die schlechte Ergebnisse bringen und trotzdem weit verbreitet sind. Eines der Bekannteren ist *Spaghetti code*, eine Beschreibung für Programmcode, der zwar tut was er soll aber dabei kaum lesbar, unverständlich und nicht zu warten ist.

Suchen Sie diesmal nach Antipatterns. Wählen Sie drei Vertreter aus und beschreiben Sie wieder jeden in drei Sätzen.

Speichern Sie Ihre Antworten in der Textdatei "aufgabe1.txt" und fügen Sie sie Ihrer Abgabe bei.

Aufgabe 2: Bilder und Model-View-Controller (10 Punkte)

Implementieren Sie eine graphische Benutzeroberfläche mit Java Swing, die folgendes leistet: Zwei Fenster werden geöffnet, in einen befinden sich zwei Knöpfe mit den Aufschriften "Erhöhen" und "Verringern", das andere ist zu Anfang leer.



Klickt man auf "Erhöhen" erhöht sich die Anzahl der Smileys, die im zweiten Fenster angezeigt werden um eins, bei "Verringern" wird sie entsprechend verringert. Maximal werden 10 Smileys in einer Reihe angezeigt (s. Bild).

Benutzen Sie die Model-View-Controller- und Observer-Entwurfsmuster um diese Anwendung zu erstellen.

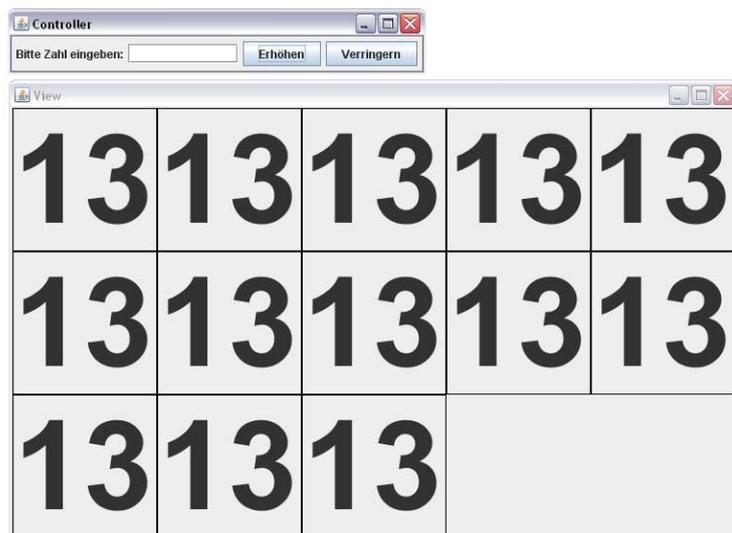
Weitere Vorgaben sind:

- Durch Klicken auf den "Schließen"-Knopf eines der beiden Fenster wird die *gesamte* Anwendung beendet.
- Weil die Anzahl der Smileys ständig variiert ist ein normales Array nicht zur Speicherung geeignet. Suchen Sie eine Datenstruktur aus dem Package `java.util`, die das flexible Hinzufügen und Entfernen von einzelnen Objekten erlaubt und verwenden Sie diese.
- Benutzen Sie als Grafik für den Smiley
<http://www.medien.ifi.lmu.de/lehre/ss08/mt/uebungen/smiley.jpg>
Laden Sie diese Bilddatei *nicht* herunter, sondern öffnen Sie sie innerhalb Ihres Programms direkt von der obigen URL aus. Erstellen Sie daraus eine Instanz der `ImageIcon` Klasse.
- Benutzen Sie als `LayoutManager` `FlowLayout`. Lassen Sie die Größe des Fensters nach dem Hinzufügen oder Entfernen eines Smileys automatisch anpassen.

Legen Sie alle so erstellten Java Quellcodedateien in einem Verzeichnis namens "aufgabe2" ab, erstellen Sie außerdem eine lauffähige JAR-Datei und fügen Sie beides Ihrer Abgabe bei.

Aufgabe 3: Geschachtelte MVC (4 Punkte)

Basierend auf der vorhergehenden Aufgabe soll eine weitere Swing-Oberfläche erstellt werden, die diesmal die eigentliche Stärke des Model-View-Controller-Musters ausspielt, indem sie mehrere verschiedene Views enthält, die alle gleichzeitig auf den neuesten Stand gebracht werden, sobald sich am Modell etwas verändert.



Dazu soll die vorhergehende Anwendung folgendermaßen erweitert werden:

- Zusätzlich zu den beiden Knöpfen wird noch ein Textfeld eingefügt mit deren Hilfe der Benutzer die Zahl im Modell direkt anpassen kann.
- Die Smileys werden durch Blöcke mit Zahlen ersetzt, die die jeweils aktuelle Anzahl von Blöcken anzeigt. Wählen Sie dafür eine geeignet große Schriftart (s. Bild). Jeder dieser Blöcke soll eine eigene View sein, und sich direkt als `Observer` beim Modell registrieren. Nur im Konstruktor der Blöcke darf der aktuelle Wert direkt aus dem Modell ausgelesen werden, ansonsten soll es über eine eigene `update`-Methode passieren.
- Maximal dürfen 25 Blöcke vorhanden sein. Die Blöcke sollen in Fünferreihen angeordnet sein. Benutzen Sie das `GridBagLayout` zur Anzeige und sorgen Sie dafür, dass nach jedem fünften Block ein Zeilenumbruch stattfindet.
- Sehen Sie sich das Video auf der Übungsseite an, um sich ein Bild der Interaktion zu machen: **<http://www.medien.ifi.lmu.de/lehre/ss08/mt/uebungen/mt1.avi>**

Legen Sie alle so erstellten Java Quellcodedateien in einem Verzeichnis namens "aufgabe3" ab, erstellen Sie außerdem eine lauffähige JAR-Datei und fügen Sie beides Ihrer Abgabe bei.

Abgabe

Packen Sie die beiden JAR-Dateien und Verzeichnisse und die Textdatei in eine ZIP-Datei und geben Sie diese Lösung bis zum 05.05.08, 10:00 Uhr im UniWorx Portal (<http://www.pst.ifi.lmu.de/uniworx>) ab.