

## Designbeschreibung

Wir erklären in den nächsten Punkten, den visuellen Aufbau des Prototypen und begründen die zentralen Designentscheidungen.

### 1. Allgemeines:

Betrachtet man sich verschiedene, dies bezieht sich auf deren Funktionalität, Programme, so fällt dem aufmerksamen Betrachter auf, dass diese Programme zumindest beim Design, also der visuellen Oberfläche, in einigen Punkten übereinstimmen. Alle Programme sind als Fenster aufgebaut. Diese Fenster sind folgendermaßen aufgebaut:

- Rahmen
- Titelleiste
- Menüleiste
- Symbolleisten
- Arbeitsfläche
- Statusleiste

- 1) Rahmen: Der Rahmen begrenzt das gesamte Fenster des Programms.
- 2) Titelleiste: Diese ist meist der obere Abschluss des Fensters. Des weiteren enthält die Titelleiste schon ein paar grundsätzliche Funktionen: Größe ändern, Fenster minimieren, maximieren und schließen.
- 3) Menüleiste: Diese ist in grundsätzliche Gruppen unterteilt: Datei, Bearbeiten, Ansicht, Hilfe; Diese Gruppen sind wieder unterteilt, so dass es leicht ist den jeweiligen Punkt zu finden.
- 4) Symbolleisten: Hier sind ausgewählte Elemente symbolisch dargestellt. Bei diesen handelt es sich meistens um die Elemente, die statistisch gesehen am häufigsten benutzt werden. Mittels des Bildes kann auch leicht erfasst werden, welche Funktion dahintersteckt.
- 5) Arbeitsfläche: In diesem Bereich läuft die hauptsächliche Benutzung des Programms ab. Er ist deswegen der visuell größte Bereich eines Fenster (ca. 95 % des Fensters).
- 6) Statusleiste: Diese Leiste befindet sich meistens unterhalb der Arbeitsfläche und bildet somit den Abschluss des Fenster. In dieser erhält der Nutzer einen groben Überblick über seine momentane Arbeit, wie zum Beispiel: Seitenanzahl, Bildanzahl, Status der Speicherung einer Datei, usw..

So in etwa ist der Grundaufbau eines Programms. Daher orientiert sich unser Programm auch an diesem grundsätzlichen Aufbau.

### 2. Produktübersicht:

Der Prototyp besteht aus 4 Fenstern. Diese besitzen alle eine Titelleiste. Die Titelleiste hat die schon oben erwähnten einstellbaren Möglichkeiten.

Das erste Fenster, auch Demo-Control Frame genannt, hat eine Arbeitsfläche, welche in 4 Buttons untergliedert ist:

1. Beenden: Durch Anklicken dieses Buttons wird das Fenster und alle anderen geöffneten geschlossen.
2. Info: Hier erhält man eine kurze Informationsangabe

3. Neues Maus Demo Panel: Durch Anklicken dieses Buttons wird ein neues Fenster, Maus Panel Demo, geöffnet.
4. Neues Koordinaten Demo Panel: Bei diesem wird ebenfalls ein neues Fenster geöffnet und zwar das Koordinatenachsen Panel Demo Fenster

Das Maus Panel Demo Fenster hat unterhalb der Titelleiste 2 Buttons. Einmal einen Schließen Button und des weiteren einen Info Button. Am unteren Rand befindet sich eine Statusleiste, die in 2 Teile unterteilt ist (linker und rechter Teil). Im linken Teil wird der Mauszustand angegeben und ob sich die Maus auf der Arbeitsfläche befindet. Im rechten Teil werden die aktuellen Koordinaten der Maus ausgegeben.

Das Koordinatenachsen Panel Demo Fenster enthält ebenfalls unter der Titelleiste einen Info und einen Schließen Button, mit ebenfalls den gleichen Funktionen wie oben schon angegeben. Unter diesen Buttons befindet sich die Arbeitsfläche. Diese hat das Aussehen eines Koordinatensystems mit x- und y-Achse. Unter der Arbeitsfläche befindet sich die Statusleiste. In dieser wird aktuelle Skalierung / Verschiebung und die Weltkoordinaten des Mittelpunkts der Arbeitsfläche angegeben.

### 3. Grundsätzliche Designentscheidungen:

Das Grundprinzip der Implementierung ist das MVC – Konzept. Also die Trennung von Modell, View und Control eines Elementes. Diesem Prinzip wurde auch durch die Bezeichnung der Klassen mit View bzw. Command Rechnung getragen. Durch dieses Prinzip wird eine strikte Trennung zwischen der graphischen Darstellung und der logischen Programmierfunktion einer Komponente verwirklicht.

Dadurch wird zu einem die Übersichtlichkeit bei großen Programmierobjekten verbessert und zum andern die Wiederverwendbarkeit der einzelnen Programmteile erhöht.

Die Klassen zur graphischen Darstellung der Fensterkomponenten wurden aus der Klasse JFrame abgeleitet, um die zahlreichen vordefinierten Methoden zu nutzen.

Eine Ausnahme bildet die InfoFensterView Klasse, welche von der Basisklasse JDialog stammt, da diese Klasse ein modales Hinweisfenster verkörpert.

Die Klassen KontrollFensterCommand, InfoFensterCommand, CPanelCommand und MPanelCommand wurden aus der Klasse AbstractAction abgeleitet, damit man auf die verschiedenen Ereignistypen der jeweiligen zugehörigen graphischen Fensterklassen reagieren kann und dadurch ist die Konstruktion komplexer Event-Handler Klassen möglich.

Damit das MVC – Konzept korrekt funktioniert müssen die jeweiligen Command-Klassen bei den entsprechenden View-Klassen registriert sein. Dies geschieht bei uns durch Übergabe der entsprechenden Command-Instanz an die bestimmte View-Instanz bei dem Aufruf des Konstruktors.

Wir haben uns entschlossen, das Koordinatensystem des Fensters Koordinatenachsen Panel durch die Methode paint zu verwirklichen. Wobei das Koordinatensystem aus einzelnen Linien gezeichnet wird. Dadurch ist es möglich damit effektiv zu arbeiten, das heißt Verschiebungen und Skalierungen durchzuführen.

#### 4. Paketstruktur und Klassenstruktur:

Wie in erstens und zweitens schon erwähnt besteht unser Prototyp aus vier Fenstern:

- Demo Kontroll Frame
- Maus Panel Demo
- Koordinatenachsel Panel Demo
- Infofenster

Jedes dieser Fenster wird durch 2 Klassen realisiert. Eine Klasse ist zuständig für die graphische Darstellung des Fensters. Diese Klassen enden jeweils mit *View* im Klassennamen. Die andere Klasse pro einzelnes Fenster ist für die Ereignisbehandlung des Fensters zuständig. Diese Klassen enden jeweils mit *Command*.

Wie in der Aufgabenstellung gefordert, existiert die Klasse Main. Diese startet das Demo Kontroll Frame.

Weiterhin existiert eine Klasse mit der Bezeichnung FensterSchliessen.Java. Diese Klasse ist für die WindowEventbehandlung zuständig, aller Fensterklassen.

Für das Applet existiert die MainApplet Klasse. In ihr wird das Applet aufgerufen und gestartet.

Alle Command Klassen sind bei ihrer jeweiligen View Klasse registriert, um die Ereignisbehandlung zu ermöglichen.

Bei Programmstart wird die Klasse Main bzw. die MainApplet aufgerufen, welche eine Instanz der Klassen KontrollFensterView und KontrollFensterCommand erzeugt. Die Klasse KontrollFensterCommand erzeugt bei Bedarf eine oder mehrere Instanzen der Klassen:

- MPanelView
- CPanelView
- MPanelCommand
- CPanelCommand
- InfoFensterView
- InfoFensterCommand

Weiterhin erzeugen folgende Klassen:

- MPanelView
- CPanelView
- InfoFensterView
- KontrollFensterView

eine Instanz der Klasse FensterSchliessen.

Hier noch eine Auflistung aller benutzen implementierten Klassen:

1. Main.java
2. Demo Kontroll Frame
  - 2.1. KontrollFensterView.java
  - 2.2. KontrollFensterCommand.java
3. Maus Panel Demo
  - 3.1. MPanelView.java
  - 3.2. MPanelCommand.java
4. Koordinatenachsen Panel Demo
  - 4.1. CPanelView.java
  - 4.2. CPanelCommand.java
5. Info Fenster
  - 5.1. InfoFensterView.java
  - 5.2. InfoFensterCommand.java
6. FensterSchliessen.java