Designbericht

Der Grundsätzlichen Aufbau eines Programms dürfte jedem Benutzer, der etwas häufiger an einem PC sitzt bekannt, sein. Wir wollen nun wie in der Aufgabenstellung gefordert, unseren Aufbau bzw. das Design etwas näher erläutern:

1. Allgemeines
2. Produktübersicht
3. Grundsätzliche Design Entscheidungen
4. Paket und Klassenstruktur

1. Allgemeines

Zuerst wollen wir den allgemeinen Aufbau beschreiben. Das Programm wird wie unter den heutigen Grafikoberflächen üblich, ein Fenstersystem besitzen. Diese Fenster sollen dem allgemeinen Standart entsprächen. Dazu gehört:

Titelleiste, Menüleiste, Symbolleiste, Buttons, Arbeitsfläche, Dialogfenster, Rahmen, Statusleiste

Titelleiste:

Ganz oben am Programmfenster zu finden. Hier steht der Name des Programms und andere für wichtig erachtete allg. Informationen über das Programm (z.Bsp. Versionsnr.). Des weiteren werden, unter vielen Betriebssystemen, 3 Buttons angeboten, Minimieren, Maximieren und Schließen.

Menüleiste:

Über die Menüleiste erreicht man die grundlegenden Programmfunktionen. Diese sind in Gruppen aufgeteilt (z.Bsp. Bearbeiten, Ansicht, usw.), die ebenfalls wieder *Unter*gruppen enthalten können.

Symbolleiste:

Enthält Verknüpfungen zu den wichtigsten Programmfunktionen über Buttons.

Buttons:

Felder mit den Programmfunktionen angesprochen werden können. Meist sind diese Funktionen durch Symbole selbsterklärend.

Arbeitsfläche:

Meist Platz mäßig größter Bestandteil eines Programmfensters, auf der die meisten Eingaben getätigt werden.

Dialogfenster:

Ein Fenster das während des Programmablaufes geöffnet wird um vordefinierte Dialoge mit dem Benutzer zu führen.

Rahmen:

Optische Begrenzung des bzw. der Programmfenster nach außen, um sie von der Restlichen Umgebung klar abzutrennen.

Statusleiste:

Befindet sich am unteren Ende eines (Dialog)Fensters. Hier werden fensterspezifische Informationen angezeigt.

2. Produktübersicht

Unserer Programm öffnet sich nach dem Start in ein Hauptfenster, von dem dann mehrere Arbeitsfenster geöffnet werden können. Das Hauptfenster besteht aus vier Buttons, über die folgende Programmfunktionen erreicht werden können: „Beenden“, „Info“rmation, „Neues MausDemoPanel“ und „Neues KoordinatenDemoPanel“. Es verfügt weiterhin, wie alle anderen Dialog- oder Unterfenster über eine Titelleiste. Die Funktionen der vier Buttons:

„Beenden“: Schließt jegliche Programmfenster, ohne weitere Aufforderung.

„Info“: Öffnet ein Fenster in dem Kurzinformation über das Programm angezeigt werden. D.h. Entwickler, Erstellungsdatum usw.

„Neues MausDemoPanel“: Es wird ein Arbeitsfenster geöffnet, in dem man mit der Maus sich bewegen kann und gleichzeitig in der Statusleiste aktuelle Informationen, über den Mauszustand und Koordinaten, ausgegeben werden.

„Neues KoordinatenDemoPanel“: Es wird ein Arbeitsfenster geöffnet, mit einem Koordinatengitter. Beim Start liegt der Koordinatenursprung zentriert. In dem Fenster soll man mit den Cursortasten sowie + und – navigieren bzw. skalieren können. Die Statuszeile enthält folgende Informationen: Skalierungsfaktor und Weltkoordinaten des Mittelpunkts.

In jedem Fenster, das unser Programm öffnet, ist ein „Beenden“- bzw. „Info“-Button verfügbar.

3. Grundsätzliche Designentscheidungen

Bei Grafikbasierten Oberflächen kann nicht mehr nach dem E*ingabe*-V*erarbeitung*-A*usgabe*-Prinzip verfahren werden und es wird ein neues Konzept benötigt. Hier kommt das MVC-Konzept zum tragen, was auch wir umgesetzt haben.

Beim MVC-Konzept wird eine Trennung in Bereiche vorgenommen:

M*odell*, V*iew*, C*ontrol*

Modell: Das M*odell* ist eine Abstraktion von reellen Vorgängen oder Systemen, das die komplette Funktionsweisen eines realen Systems darstellt. Das M*odell* ist somit nicht nur ein Container für Daten, sondern auch eine Sammlung von Algorithmen um diese Daten zu verarbeiten.

View: Der V*iewport* ist die grafische Schnittstelle des Programms, also eine Sicht auf ein M*odell*. Ein *Modell* kann mehr als nur ein V*iewport* haben, um die Daten aus verschiedenen Sichtweisen besser sehen zu können. Ändern sich die Daten in einem M*odell*, so wird der V*iewport* benachrichtigt und aktualisiert seine Ausgabe.

Control: Der *Controller* ist der Eingabeteil beim MVC-Konzept. Durch ihn bekommen die *Modelle* und die *Viewports* Instruktionen vom Anwender und können sich dem entsprechend verhalten. Um die verschiedenen Eingabemöglichkeiten nicht fest mit dem Algorithmen zu koppelt wird dieses in den C*ontroller* ausgelagert, um ihn später leichter an Veränderungen anpassen zu können.

Um die Methoden die Java zur graphischen Darstellung bereitstellt nutzen zu können, haben wir die Klasse JFrame abgeleitet, um unsere Viewports zu realisieren. Eine Ausnahme bildet das Info Fenster, was von JDialog abgeleitet ist.

Die Controller nutzen Methoden von AbstractAction. Diese vordefinierten Methoden wurden wieder mittels Vererbung genutzt, um die Eventhandler für die Viewports, sprich Fenster, zu implementieren.

4. Paket- und Klassenstruktur

Wie oben erwähnt haben wir nach dem MVC-Konzept gearbeitet und die in der Aufgabenstellung geforderten Fenster wie folgt gegliedert:

- Hauptfenster bzw. DemoKontrollPanel
- Mausfenster bzw. MausDemoPanel
- Koordinatenfeld bzw. KoordinatenDemoPanel
- Infodialog bzw. Info

Beim Start des Programms wird in die Main-Klasse gegangen. Diese öffnet das DemoKontrollPanel. Von dort werden die geforderten Klassen MPanel- und CPanelView bzw. InteractioncontrolArea angesprochen. Die ~View implementieren, wie der Name vielleicht schon sagt, die Viewports und die Interaction verkörpert den Controller, der die Benutzereingaben an die Viewports und die Modells weitergibt.
Um das Applet zu starten existiert die Klasse AppletView. Die InteractionArea-Klasse bilde

Wird nun das Programm aufgerufen, so wird aus der Klasse AppletView oder Main eine oder mehrere Instanzen der Control und View Klassen erzeugt.

Klassenübersicht

(i) Main (DemoKontrollPanel)
(ii) MpanelView
(iii) CpanelView
(iv) InteractionArea
(v) CordinateArea (CPanelViewer Koordinatensystem)
(vi) CordinateArea2 (MPanelViewer Koordinatensystem)
(vii) InfoBox