

GE001 - SOFTWARE SOLUTIONS

04109 Leipzig • Augustusplatz 10
mail: softwaresolutions@everyday.com

Erstellt von: Anne Güpner, Dominic Rose, Marcel Kretschmann
am: 01.05.03



Designbeschreibung

1. Allgemeines

Im Prototyp werden allgemeine Entwicklungs-Konzepte umgesetzt. GUI-Programmierung und Ereignisbehandlung werden unter Berücksichtigung des MVC – Ansatzes (Model – View – Control) vollzogen. Der Prototyp dient der Einarbeitung und vorläufigen Umsetzung dieser Ansätze.

Für die Ausführung des Programms sind Java SDK bzw. JVM nötig. Mit dem Internetexplorer und jedem anderen Browser kann das Applet mit Hilfe entsprechender Java-Plugins problemlos ausgeführt werden. Auch als Applikation ist der Prototyp ausführbar.

2. Produktübersicht

Der Prototyp besteht aus einem Kontrollfenster von dem aus sich drei Weitere mittels Buttons öffnen lassen. Alle 4 Buttons haben eine spezifische Funktion. Drückt man den Button mit der Aufschrift „Beenden“ so schließen sich alle noch offenen Fenster und die Applikation selbst. Bei den anderen drei Knöpfen öffnen sich neue Fenster, wobei das Kontrollfenster jeweils offen bleibt. Eine kleine Info erscheint, wenn man den Button „Info“ drückt. Das Fenster „Neues MausDemoPanel“ dient zur Demonstration der Mausereignisbehandlung. Innerhalb des neuen Fensters wird in einer Statuszeile die aktuelle Position des Mauszeigers angezeigt, sofern er sich innerhalb des Fenster befindet. Außerdem wird ermittelt, on der Mauszeiger gerade über einer bestimmten Komponente ist. Der Button „Neues KoordinatenDemoPanel“ öffnet ein Fenster in welchem ein Koordinatengitter zu sehen ist. Das Koordinatengitter hat seinen Ursprungspunkt beim Öffnen des Fensters in der Mitte. Nimmt man nun die Kursertasten (oder die Tasten + und –) so kann man den in der Zeichenfläche dargestellten Ausschnitt verschieben (oder skalieren). Diese Verschiebung wird wiederum in einer Statuszeile festgehalten, dabei werden der Zoomfaktor und die Koordinaten des Ursprungs angezeigt. Die beiden letzten Fenster enthalten jeweils einen Knopf zum „Schließen“ und einen, bei welchem eine kleine „Info“ angezeigt wird.

3. Grundsätzliche Design Entscheidungen

Da das MVC Konzept auf einer Dreiteilung des Programmcodes nach seiner Funktionalität beruht, musste die Klassenwahl nach dem Prinzip des MVC-Kozeptes erfolgen.

Um diese Teilung in Modell, Controller, und View vorzunehmen wurden jede Klassen (außer der Klasse Hauptklasse, die die Methoden main, bzw. init enthält) einem der drei Bereiche zugeordnet.

Es gibt nun drei Arten von Klassen:

Die Modell-Klassen sind die Klassen die für die Datenspeicherung und -berechnung zuständig sind.

Namentlich ist das die Klasse Modellklasse und die nur für die Modellklasse relevante Klasse ViewListElement (welche nur dazu dient um eine Liste, aller zur Modellklasse gehörenden ViewObjekte, zu erstellen und auf diese zugreifen zu können).

Die Controller-Klassen erzeugen alle Fenster und Schaltelemente, ebenso registrieren und Verarbeiten sie jedwede Aktion des Nutzers mit der Nutzeroberfläche des Programms. Diese Klassen schicken die Eingaben die der Benutzer gemacht hat an das Modell welche sie weiterverarbeitet.

Zu diesem Zweck gibt es eine Basisklasse Controller, von der sich alle anderen Controller-Klassen ableiten. Es gibt um die verschiedenen Funktionalitäten des Programms zu gewährleisten wurden insgesamt drei Controller-Klassen von Controller abgeleitet.

Die HauptController-Klasse, diese Klasse verwaltet die Eingaben für das Hauptfenster, welches die anderen Fenster aufrufen kann.

Die MausController-Klasse, welche zur abfrage der Mausfunktionalität im Maus-Demo-Fenster dient.

GEO01 - SOFTWARE SOLUTIONS

04109 Leipzig • Augustusplatz 10
mail: softwaresolutions@everyday.com

Erstellt von: Anne Güpner, Dominic Rose, Marcel Kretschmann
am: 01.05.03



Die TastenController-Klasse, welche die Tastatursteuerung und Eingabefunktionalität im Koordinaten-Fenster des Programmes sichert.

Zu guter letzt wurden noch die View-Klassen angelegt, welche zur Steuerung der Ausgabe des Programmes dienen. Diese Klassen erhalten ihre Daten, anhand denen die Ausgabe für den Nutzer erstellt wird.

Es gibt eine Basis-View-Klasse, die die grundlegende Funktionalität, wie das Anmelden beim Modell, damit dieses die Nachricht zur Datenerneuerung schicken kann, enthält.

Von dieser Basisklasse wurden zwei View-Klassen abgeleitet, die jeweils bei den bei dem Maus-Demo-Fenster und bei dem Koordinaten-Fenster eingesetzt werden.

Zum einen die MausView-Klasse, welche dem Benutzer die wichtigsten Mauszustände angibt.

Und zum anderen die KoordinatenView-Klasse, welche das Koordinatensystem in Abhängigkeit der im Modell vorhandenen Daten anzeigt.

Die MausController-Klasse und die MausView-Klasse wurden einander zugeordnet.

Die TastenController-Klasse und die KoordinatenView-Klasse wurden einander zugeordnet.

Es gibt keinen View für die HauptController-Klasse, da hier keine Daten für den Nutzer dargestellt werden müssen.

Die Basisklassen View und Controller sind, aufgrund ihres Funktionalitätsmangels, abstrakte Klassen und können nicht instanziiert werden.

4. Paket- und Klassenstrukturen

Die Paketstruktur des Prototyps ist recht simpel: „Default“ = alle 4 Dateien in ein Verzeichnis...

Darstellung der verwendeten Klassen als Klassendiagramm mit Hilfe des CASE Tools Rational Rose:

GEO01 - SOFTWARE SOLUTIONS

04109 Leipzig • Augustusplatz 10
mail: softwaresolutions@everyday.com

Erstellt von: Anne Güpner, Dominic Rose, Marcel Kretschmann
am: 01.05.03

