

Designbeschreibung von GEO05

Version 1.1

Verantwortlicher: Raik Quakatz
Martin Wannagat

1. Allgemeines:

Die Java – Applikation GEO05 ist eine menügesteuerte graphische Anwendung, mit deren Hilfe sich geometrische Konfigurationen visualisieren lassen.

Als Voraussetzung zum starten dieser eigenständigen Applikationen benötigt man JAVA JRE 1.4. GEO05 verfügt zur Bedienung über eine grafische Nutzerschnittstelle mit allgemein üblichen Fenster-techniken.

2. Produktübersicht:

GEO05 wird durch den Kommandozeilenaufruf

```
java -jar geometry.jar
```

im Wurzelverzeichnis der Pakethierarchie gestartet. Wenn man über Java 1.4.1_01 oder höher verfügt kann die Applikation auch durch doppelklicken der *geometrie.jar* Datei gestartet werden.

Wenn dieser Kommandozeilenaufruf erfolgt ist wird das Hauptfenster initialisiert.

Das Anwendungsfenster in GEO05 ist, von oben nach unten, in folgende Komponente untergliedert:

- Titelleiste
- Menüleiste
- Symbolleisten
- Desktop-Fläche
- Statusleiste

Titelleiste: Diese bildet den oberen Abschluß des Fensters. Sie enthält, neben dem Namen der Applikation, grundsätzliche Fensterfunktionen, wie die Minimierung, die Maximierung oder die Terminierung des Fensters.

Menüleiste: Diese ist in grundsätzliche Gruppen unterteilt: Datei, Projekt, „?“; Diese Menüeinträge sind wiederum unterteilt, so das eine intuitive Zuordnung der Applikationsfunktionen gegeben ist.

Symbolleisten: Hier sind ausgewählte Elemente symbolisch dargestellt. Bei diesen handelt es sich zumeist um die Elemente, die statistisch gesehen am häufigsten benutzt werden. Mittels der Icon-piktogramme ist eine Funktionsdefinition der Symbole gegeben.

Desktop-Fläche: In diese Fläche lassen sich im weiteren Verlauf die verschiedenen Zeichenflächen sowie ein spezielles Fenster für Fehlermeldungen öffnen.

Statusleiste: Diese Leiste befindet sich am unteren Rand und bildet somit den Abschluss des Fenster. In dieser erhält der Nutzer einen groben Überblick über seine momentane Arbeit.

Nach dem Starten des Programms kann man in der Menüleiste über den Menüeintrag *Datei* und dem Untermenüpunkt *Öffnen* ein vorhandenes Projekt öffnen, bzw. durch Untermenüpunkt *Neu* ein neues Projekt anlegen.

Es ist möglich mehrere Projekte zu öffnen. Diese verschiedenen Fenstern der Desktopfläche können nun abwechseln, mit der Maus, aktiviert werden.

Alternativ kann das aktuelle Fenster auch über den Menüeintrag *Window* gewechselt werden.

Parameterwerte eines aktivierten Fensters(aktivierte Zeichenfläche) können in einem Dialogfenster modifiziert werden. Die Visualisierung wird neu dargestellt.

Die aktiven Fenster kann über einen weiteren Menüpunkt oder einen speziellen Knopf am Fensterrand geschlossen werden.

Die Applikation kann dann anschließend über den Exit-Eintrag im File -Menü beendet, oder dem Icon in der Titelleiste beendet werden.

3. Grundsätzliche Designentscheidungen

Dieses Geometrieprogramm ist in der Programmiersprache Java implementiert wodurch das Programm plattformunabhängig ist.

Das Grundprinzip der Implementierung ist das MVC – Konzept, also die Trennung von Modell, View und Control eines Elementes.

Die graphische Oberfläche von GEO05 wird mit Swing-Klassen, gängigen Fenster-Techniken und dem Ereignis-Aktions-Konzept realisiert.

In dem Paket „Command“ werden die Informationen über die verschiedenen geometrischen Objekte zusammengefaßt. Dabei wird zunächst eine allgemeine Oberklasse „GeoElement“ angelegt. Von dieser werden dann die speziellen geometrischen Elemente wie Kreis, Punkt, Strecke usw. abgeleitet. Dabei dienen diese Klassen nur zur Zusammenfassung der Kenngrößen der geometrischen Objekte, aber sie beinhalten keine Funktionalität zur graphischen Darstellung dieser Objekte.

In dem Paket „view“ sind die Klassen abgelegt, welche für die Darstellung der graphischen Benutzeroberfläche zuständig sind. So findet man hier zum Beispiel die Klasse AnwendungFenster, welche das Hauptfenster der Applikation erzeugt. Diese Klasse wurde aus der vordefinierten Klasse JFrame abgeleitet, damit alle vorgefertigten Methoden für ein typisches Fenster wiederverwendet werden können.

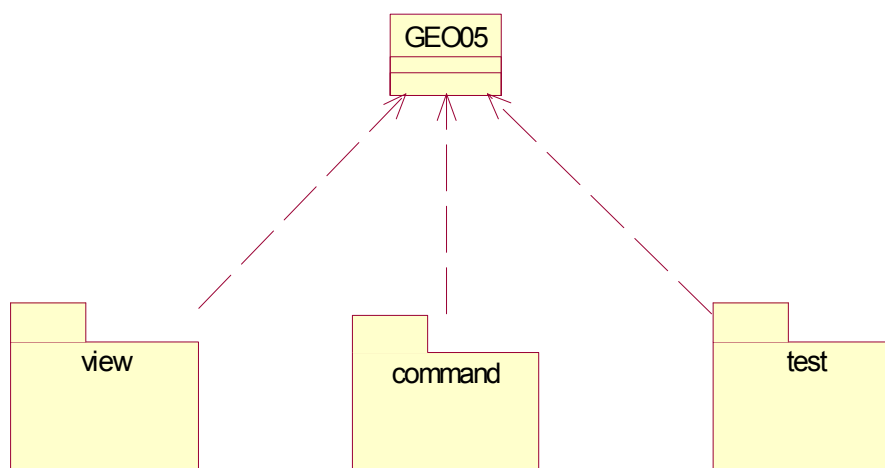
In diesem Paket sind aber auch die Klassen angeordnet, welche für die Ereignisbehandlung der graphischen Oberfläche zuständig sind. So ist zum Beispiel die Klasse *Fenster schliessen* dafür da, um auf `exitActions` zu reagieren, also wenn jemand ein Fenster schließen will, dann muss diese Klasse darauf mit einer Action reagieren.

Es wird eine Trennung zwischen der graphischen Darstellung der Elemente, wie Fenster, geometrischer Objekte und den programmlogischen Funktionen dieser Elemente durchführt. Damit wird dem MVC-Konzept Rechnung getragen. Dies dient vor allem dazu, die Übersichtlichkeit bei mittleren und großen Softwareprojekten zu wahren. Aber auch die Wiederverwendbarkeit des Quellcodes wird dadurch erhöht, denn man kann nun beispielsweise die selbe graphische Darstellung mit komplett anderen programmlogischen Funktionen ausstatten bzw. die vorhandenen Funktionen sehr leicht erweitern.

4. Paket- und Klassenstruktur

Zunächst betrachten wir das statische Modell. Des weiteren wird vorerst noch ein Anwendungsfall zur Darstellung des dynamischen Modells der Applikation gezeigt.

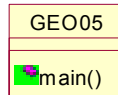
Die Applikation ist in drei Pakete und eine Hauptklasse aufgeteilt.



4.1. Die Hauptklasse GEO05

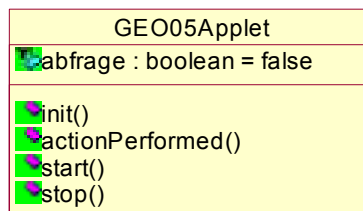
Die Hauptklasse GEO05, die auch die main-Methode enthält, ist für die Initialisierung der Arbeitsumgebung (Hauptbildschirm) verantwortlich.

Im Erfolgsfall wird der Hauptbildschirm, eine Instanz von `view.Anwendungsfenster`, aufgebaut und die Steuerung dorthin übergeben.



GEO05 kann ebenfalls als Applet gestartet werden.

Dies geschieht durch die Klasse GEO05APPLET. Es wird zur ContentPane ein JButton hinzugefügt und mittels ActionListener abgefragt ob dieser Button gedrückt wurde. Ist dies der Fall wird das Applet gestartet.



4.2. Das Paket view

Dieses Paket enthält alle Fenster- und Action-Klassen sowie die Klassen, welche die Visualisierung geometrischer Objekte steuern.

Die **zentrale Klasse** ist Anwendungsfenster, welche die Swing-Klasse JFrame erweitert und von der im Laufe des Programms eine Instanz, das Hauptfenster der Applikation, angelegt wird.

Die einzelnen Menüs bzw. Menüpunkte werden direkt als Instanzen von JMenu oder JMenuItem angelegt und im Konstruktor Anwendungsfenster() initialisiert. Nach dem Event- Action-Konzept gibt es zu jeder Aktion eine Aktionsklasse:

- Projekt öffnen
- Programm schließen

, welche die Swingklasse Abstract- Action erweitert.

Die Verwaltung der einzelnen Zeichenflächen erfolgt in einer Instanz der Klasse Hintergrund, die als Erweiterung der Swingklasse JInternalFrame für den Hintergrund desjenigen Bereichs im Hauptbildschirm steht, in dem die Zeichenflächen dargestellt werden. Es können mehrere Projektfenster geöffnet sein, wobei jedoch nur eines den Status aktiv besitzt.

Die Statuszeile am unteren Rand des Projektfensters wird als Instanz von StatusBar angelegt.

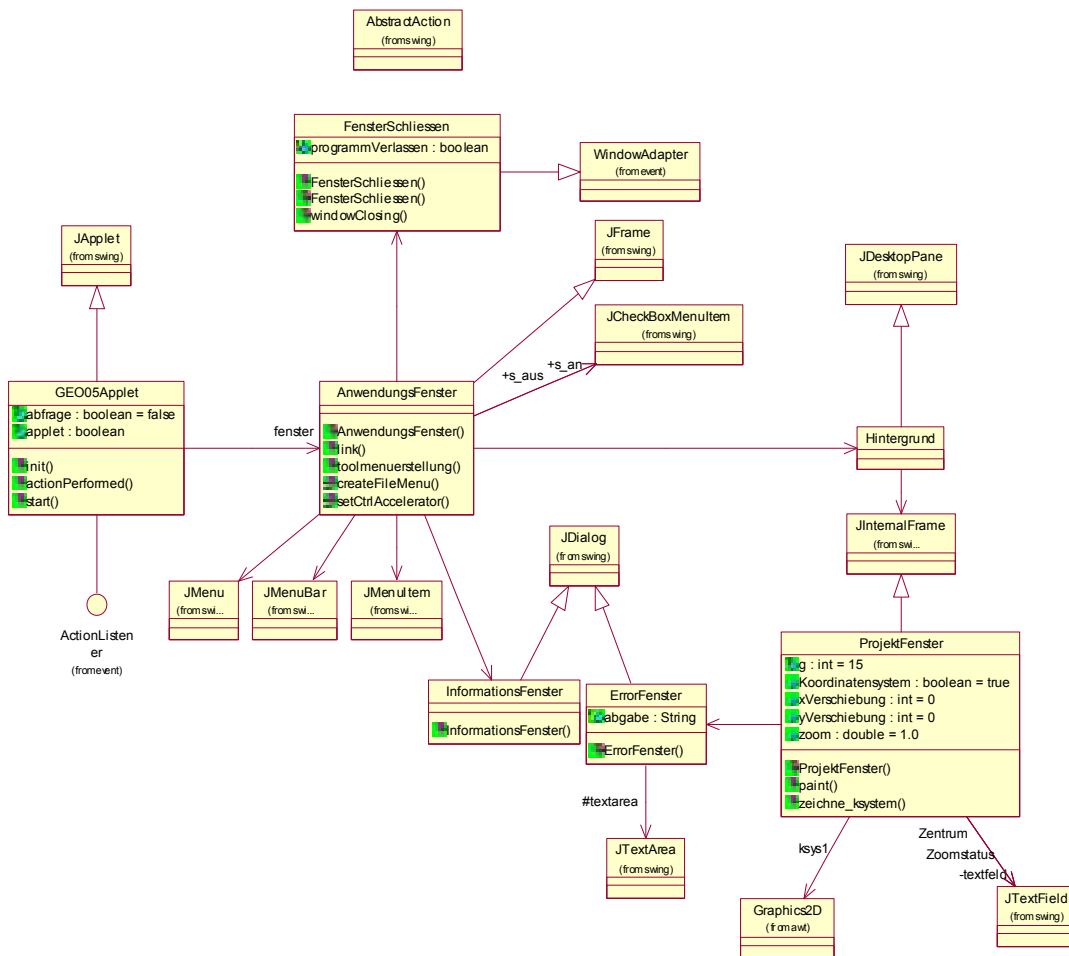


Abb. 1: Paket view

Ein GeoPoint besteht aus (x,y)- Koordinaten.

Eine GeoLine wird mit zwei Integerwerten charakterisiert. Jeder Wert gibt eine Position eines Punktes in der Punktliste an.





Ein GeoCircle wird mit einem Integerwert und einem Doublewert charakterisiert. Der Integerwert ist wieder eine Position eines Punktes in der Punktliste. Der Doublewert gibt den Radius des Kreises an.

Ein SGeoPoint besteht aus zwei Linienpositionen der Linienliste und einer Punktposition in der Punktliste.

Ein MGeoPoint besteht aus einer Linienposition und einer Punktposition.

Ein GeoGleiter besteht aus einer Linienposition und einer Punktposition sowie dem Abstand zu Punkt 1 der Linie.

4.4 Das Paket test

Tests
 testMittelpunktBerechnung()
 testeinfuegenUNDloeschen()
 testKreis()
 testSchnittpunktBerechnung()

In dieser Klasse werden die Mittelpunkt-, Schnittpunktberechnung überprüft. Außerdem wird getestet, ob alle Einträge der Listen nach Einfügen bzw. Löschen von Elementen korrekt sind.

In testKreis wird überprüft, ob ein Mittelpunkt und Radius richtig in Kreisliste eingefügt wird.

Im weiteren folgt nun eine Übersicht über die Klassen des view-Paketes sowie des Command-Paketes, die die Gruppe GEO05 im Laufe der Implementierungsphase bereits implementiert hat.

Die Klasse *Anwendungsfenster* welche sich, wie aus dem Namen ersichtlich ist, im View-Paket befindet, legt das Layout des Anwendungsfensters fest. Es wird als erstes eine Instanz *hintergrund* von *JDesktopPane* angelegt. Zudem wird der *Desktopmanager* initialisiert, welcher die Interalfames verwaltet.

Es wird eine Hintergrundfarbe festgelegt und der Hintergrund initialisiert. Im weiteren wird das Menü angelegt und jedem Menüpunkt ein Abhörer zugeordnet.

Die Klasse *AnwendungsfensterCommand* ist dem Command-Paket zuzuordnen. Sie ordnet der Variablen Quellfenster das aktive Fenster, in welcher eine Aktion stattfindet, zu. In der Methode *actionPerformed* werden die Reaktionen auf die Aktionen festgelegt.

Die Klasse *ErrorFenster* erzeugt ein Fenster zur Ausgabe eines Fehlertextes.

Die Klasse *ErrorFensterCommand* ist Bestandteil des Command-Paketes und stellt Reaktionen auf Aktionen im Errorfenster bereit.

Diese Klasse *InformationsFenster* welche sich, wie aus dem Namen ersichtlich ist, im View-Paket befindet, legt das Layout des Informationsfensters fest. Es wird ein Fenster vom Typ *JDialog* erzeugt welches Labels und ein OK-Button enthält. Zusätzlich wird ein Abhörer für den OK-Button initialisiert.

Diese Klasse *InformationsFensterCommand* ist, wie der Name sagt, Bestandteil des Command-Paketes und stellt Reaktionen auf Aktionen im Informationsfenster bereit, so auch das schließen des Fensters durch drücken auf die Return-Taste

Die Klasse *FensterSchliessen* welche trotz nicht eindeutiger Kennzeichnung zum Controller-Paket gehört, liefert die Methoden zum terminieren des Anwendungsfenster und vorher natürlich das schliessen sämtlicher Projektfenster

Die Klassen Laden und Speichern stellen die graphische Oberfläche für die Dateiauswahl zum Laden und Speichern einer Datei bereit.

Die Klasse *Projektfenster* welche sich, wie aus dem Namen ersichtlich ist, im View-Paket befindet, legt das Layout der Projektfenster fest. Zusätzlich wird ein Abhörer für das Fenster initialisiert. Es wird ein Fenster vom Typ *JInternalFrame* erzeugt, in welchen Interaktionen stattfinden können.

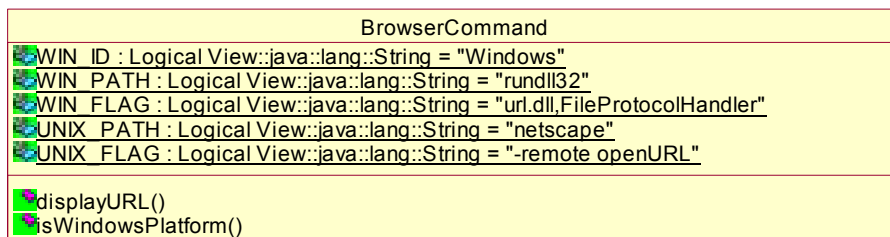
Die Klasse liefert ausserdem die X- und Y-Koordinaten der Mausklicks(zum Beispiel beim Punkt zeichnen) an die Klasse *GeoPoint* („Controller“), welche dann wiederum den Punkt zeichnet.

Die Klasse *ProjektFensterCommand* welche sich, wie aus dem Namen ersichtlich ist, im Controller-Paket befindet, fängt die Aktion des Projektfenster auf und liefert die entsprechenden Reaktionen.

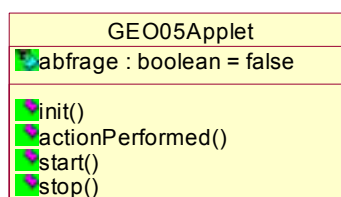
BrowserFensterCommand:

Diese Klasse dient zur Darstellung der Hilfedateien. Die Hilfedateien werden mit Hilfe des installierten Browsers aufgerufen.

Mit dieser Klasse ist es möglich herauszufinden unter welchem Betriebssystem die Applikation läuft. Dadurch ist eine gute Interaktion zwischen der Hilfe und dem Betriebssystems gewährleistet.



Die Klasse *GEO05Applet* wird benötigt um das Applet zu starten. Es wird zur *ContentPane* ein *JButton* hinzugefügt und mittels *ActionListener* abgefragt ob dieser Button gedrückt wurde. Ist dies der Fall wird das Applet gestartet



4.4 Dynamisches Modell

Das dynamische Modell wird an zwei ausgewählten Anwendungsfällen der Version 1(Pflichtenheft) veranschaulicht.

4.4.1 /F10/

In diesem Anwendungsfall wird das Öffnen eines neuen Projektes dargestellt.

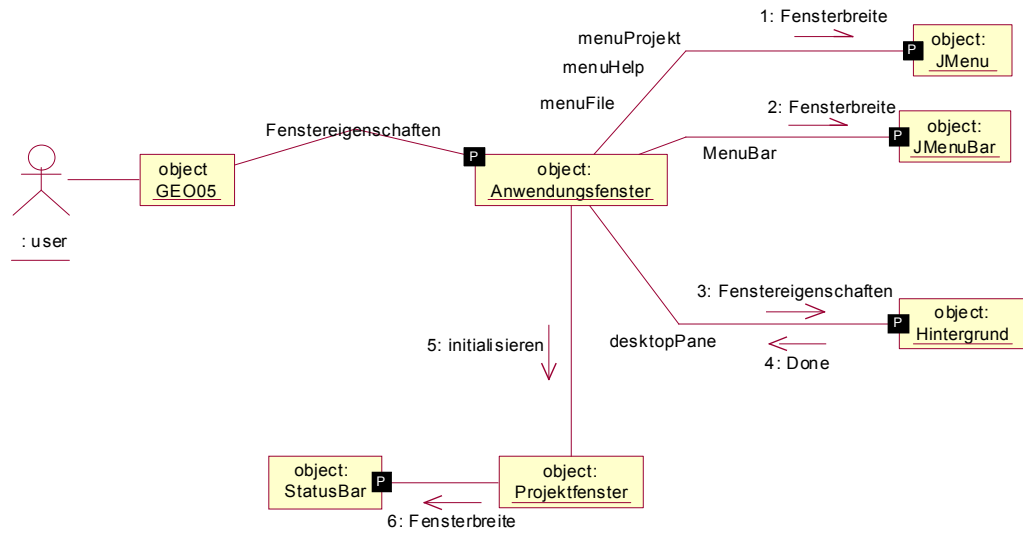


Abb. 5: Kollaborationsdiagramm /F10/

Nachdem das Anwendungsfenster(mit Jmenu, JmenuBar und Hintergrund) vollständig aufgebaut ist, kann die Funktion /F10/ aus dem Menü gewählt werden. Es wird ein neues Zeichenfenster(Projectfenster) mit StatusBar angelegt.

Im folgendem wird ein Sequenzdiagramm abgebildet, um /F10/ auch zeitlich näher zu spezifizieren, d.h. es wird der zeitliche Ablauf von /F10/ dargestellt.

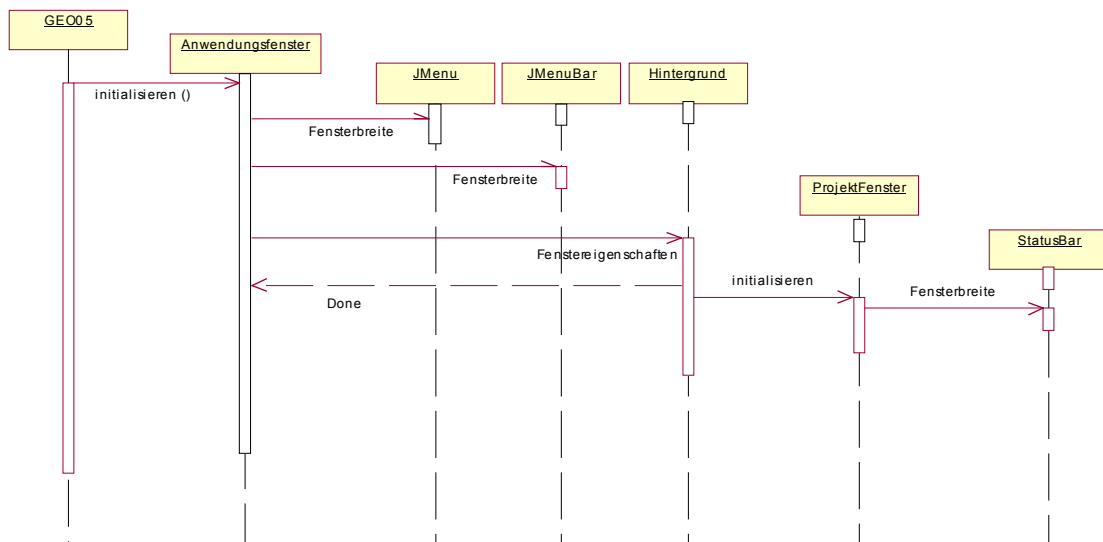


Abb. 6: Sequenzdiagramm /F10/