Design-Beschreibung der Applikation Geo09

Version 1 von 08.06.03

Gruppe Geo09 **Design- Verantwortlicher**: R. Hrushchak

Erstellt von: M. Chen, R. Hrushchak, P. Kongsto, A. Stock

1. Allgemeines:

Kurzcharakterisierung:

Die Software dient zur Durchführung und späteren dynamischen Veränderung von geometrischen Konstruktionen.
Sie wird in Java geschrieben, verfügt über eine graphische Oberfläche und wird durch Menü bzw. Maus gesteuert. Die Applikation kann auch als Applet innerhalb eines Browsers gestartet werden.

Systemvoraussetzungen:

Voraussetzungen zum Starten: Java JRE 1.4.
Für eine zügige Arbeit wird folgende minimale Rechnerkonfiguration empfohlen:

- Prozessor: P2 (300 Mhz),
- Arbeitsspeicher: 64 MB,
- Monitor mit Auslösung 800 x 600 Pixel

Abgrenzung

Die erste Version der Software verfügt nicht über folgende Funktionalitäten:

- Speichern
- Exportieren in andere Datei- Formate
- Benutzerdefinierte Einstellungen
- Drucken

2. Produktübersicht

Die Anwendung wird durch Kommandozeilen-Aufruf gestartet, wird aber nur durch GUI bedient.
Das Hauptfenster enthält am oberen Rand einen **Menü**- und **Toolbalken**, am unteren Rand einen **Statusbalken** und in der Mitte eine **Desktop-Fläche**, in der im weiteren Verlauf neue Zeichenflächen geöffnet werden können.
Das **Menü** besteht aus folgenden Einträgen:

Menüeinträge	***Übersicht***
Datei: • Neue Konstruktion (*) • Neue Zeichenfläche • Konstruktion schließen • Zeichenfläche schließen • Beenden	Es ist möglich, mehrere Konstruktionen zu öffnene bzw. zu der aktuellen Konstruktion neue Zeichenflächen zu öffnen. Wenn die aktuelle Konstruktion geschlossen wird, verschwinden auch alle, zu ihr zugehörige, Zeichenflächen. Beim Schließen einer Zeichenfläche verschwindet nur die aktuelle Zeichenfläche, die entsprechende Konstruktion sowie die anderen Zeichenflächen-Schwestern bleiben offen. Beim Beenden der Anwendung werden alle geöffneten Konstruktionen und Zeichenflächen geschlossen.
Bearbeiten: • Rückgängig (*) • Wiederherstellen (*)	Die Anzahl der Operationen, die man rückgängig machen bzw. wiederherstellen kann, ist auf 10 begrenzt.

Menüeinträge	*Übersicht*
Ansicht: • Vergrößern + (*) • Verkleinern - (*) • Zoom 100 % • Sichtbaren Bereich Verschieben	Der aktuellen Zeichenfläche kann ein Scalierungsfaktor zugewiesen werden. Dies erfolgt durch Anklicken des Befehles „Vergrößern“ bzw. „Verkleinern“. Zur Standard-Darstellung kehrt man mit dem Befehl „Zoom 100 %“ zurück. Nach das Anklicken des Befehles „Sichtbaren Bereich verschieben“ ist es möglich, den sichtbaren Bereich der Zeichenfläche mit der Maus zu verschieben.
Zeichenfläche: • Koordinatensysem ein- / ausblenden (*) • Sichtbaren Bereich Verschieben • Gitter ein- /ausblenden (*)	Koordinatensysem bzw. Gitter wird nach dem Anklicken des entsprechenden Befehles ein- / ausgeblendet.

Objekte:
• Punkte:
  • Punkt (*)
  • Gleiter (*)
  • Schnittpunkt (*)
  • Mittelpunkt (*)
• Geraden:
  • Gerade (*)
  • Strecke (*)
  • Lot (*)
• Gruppe
  • bilden + (*)
  • auflösen - (*)
  • Element ausgruppieren
• Bewegen (*)
• Objekteigenschaften
	Nach dem Anklicken eines der Befehle ist man in der Lage, links genannte geometrische Objekte zu erstellen.

Fenster: • < Liste aller geöffneten Zeichenflächen >	Um zwischen den geöffneten Fenstern zu wechseln, muss man den entsprechnden Namen unter dem Menüeintrag „Fenster“ anklicken.
Hilfe: • Hilfe • Information	Information und Hilfe zur Software bekommt man im neuen Fenster nach Anklicken des Befehles „Hilfe“ bzw. „Information“.

Funktionen, die in der Menü-Tabelle mit einem Sternchen (*) bezeichnet sind, bilden eine Menge von Buttons, aus welchen der **Toolbalken** besteht**.** Die **Buttons** werden als Icons, ohne Beschriftung, dargestellt.

Aktueller Zustand (z.B. Koordinaten eines angeklickten Punktes) werden am unteren Rand auf einem **Statusbalken** eingeblendet.

3.Grundsätzliche Design-Entscheidungen

Programmablauf

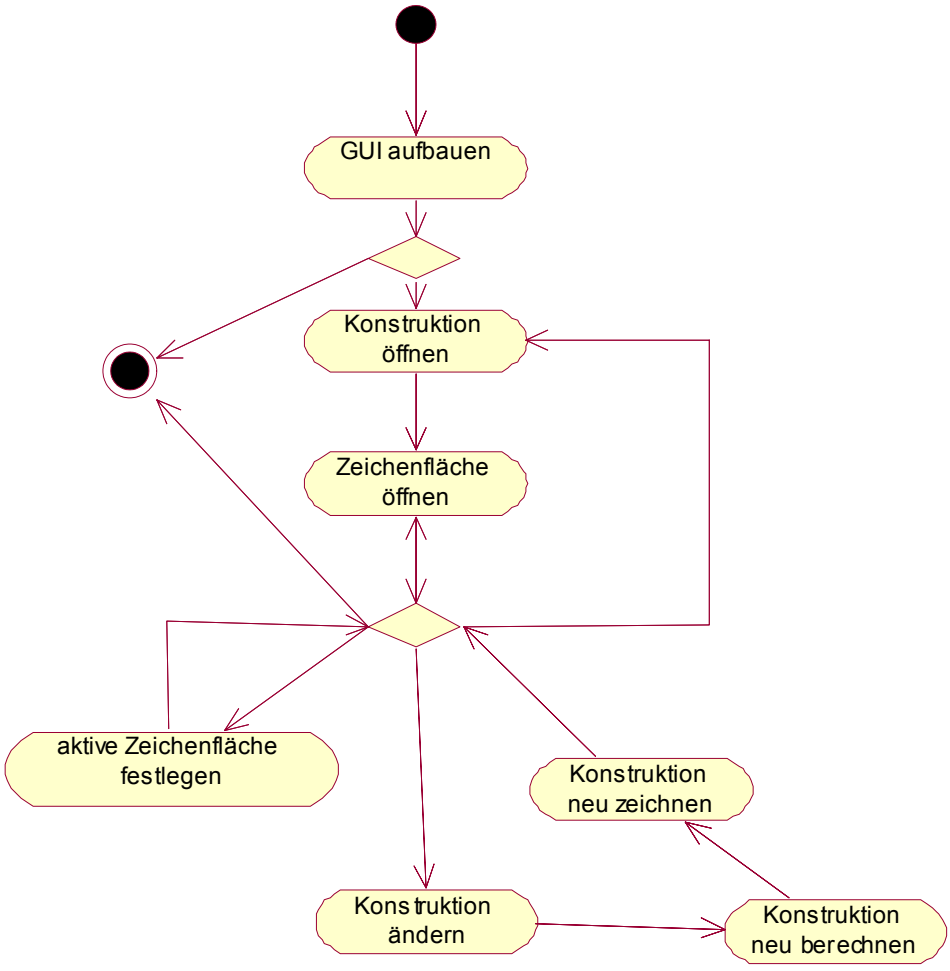

Diagramm 1

Nach dem Starten der Anwendung wird GUI aufgebaut. Die Software kann aber zu jeder Zeit geschloßen werden (Übergang zum Endzustand).
Man kann auch immer eine neue Konstruktion bzw. Zeichenfläche öffnen.
Außerdem ist es möglich, zwischen geöffneten Zeichenflächen zu wechseln und somit die aktive Zeichenfläche festzulegen.
Auf der aktiven Zeichenfläche kann man geometrische Konstruktionen bilden, ändern und bewegen. Nach jedem Vorgang werdes geänderte Attribute des Modelles neuberechnbet und neugezeichnet.

Programmzustände

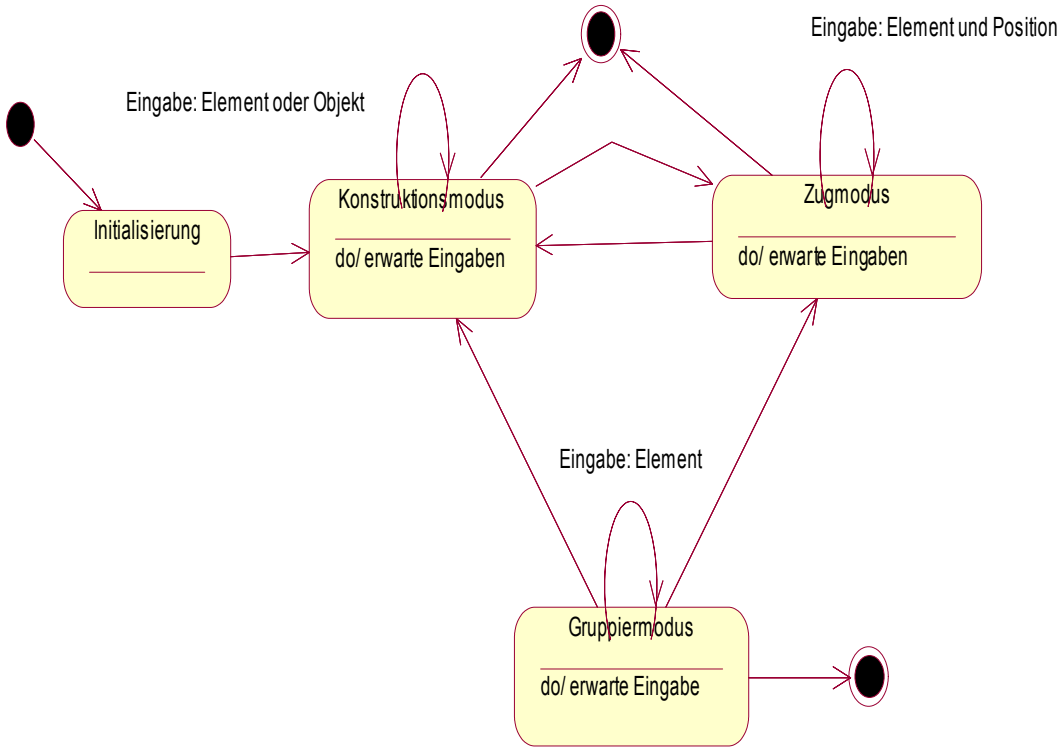

Diagramm 2

Nach der Initialisierung befindet sich das Programm sofort im Konstruktionsmodus. Dieser ist ein rein logischer Modus, der zur Vereinfachung der Erklärung hier eine Menge von Zuständen repräsentiert, z.B. Gerade erstellen, Schnittpunkt berechnen, freien Punkt hinzufügen usw. Allen diesen ist gemeinsam, dass das Programm eine bestimmte Art von Nutzereingaben erwartet, für eine Gerade z.B. die Auswahl zweier Elemente aus einer Ansicht. Sind diese ausgewählt, bleibt das Programm in Modus „Gerade erstellen“ und erwartet wiederum die Auswahl zweier Elemente. Mittels entsprechender Nutzereingaben, z.B. „Gruppe erstellen“ bzw. „Element bewegen“, wird in den Gruppier- bzw. Zugmodus übergegangen. Diese erwarten die entsprechenden Eingaben (Elemente bzw. Elemente und Positionen). Der Wechsel zwischen diesen drei Modi, die für den Nutzer nicht direkt sichtbar sind, ist jederzeit möglich. Die Transitionen erfolgen durch Betätigen eines mit diesen Modi assoziierten Buttons, andere Übergänge zwischen diesen Zuständen finden nicht statt. Die auf Eingaben folgende Programmaktivität hängt natürlich vom aktuellen Zustand ab – das Auswählen eines Punktes erfordert im Gruppiermodus anderes Verhalten als im Konstruktionsmodus.

Selbiger ist also als eine Zusammenfassung mehrerer sehr spezieller Modi zu verstehen, wobei Untermodi zu jeder Funktion, welche das Hinzufügen von Elementen oder Objekten zu Konstruktionen beinhaltet, existieren.

Als Beispiel zeigt folgendes Diagramm das Erstellen einer Gerade, wobei Start- und Endzustand andere Unterzustände des Konstruktionsmodus, den Gruppier- oder Zugmodus darstellen.

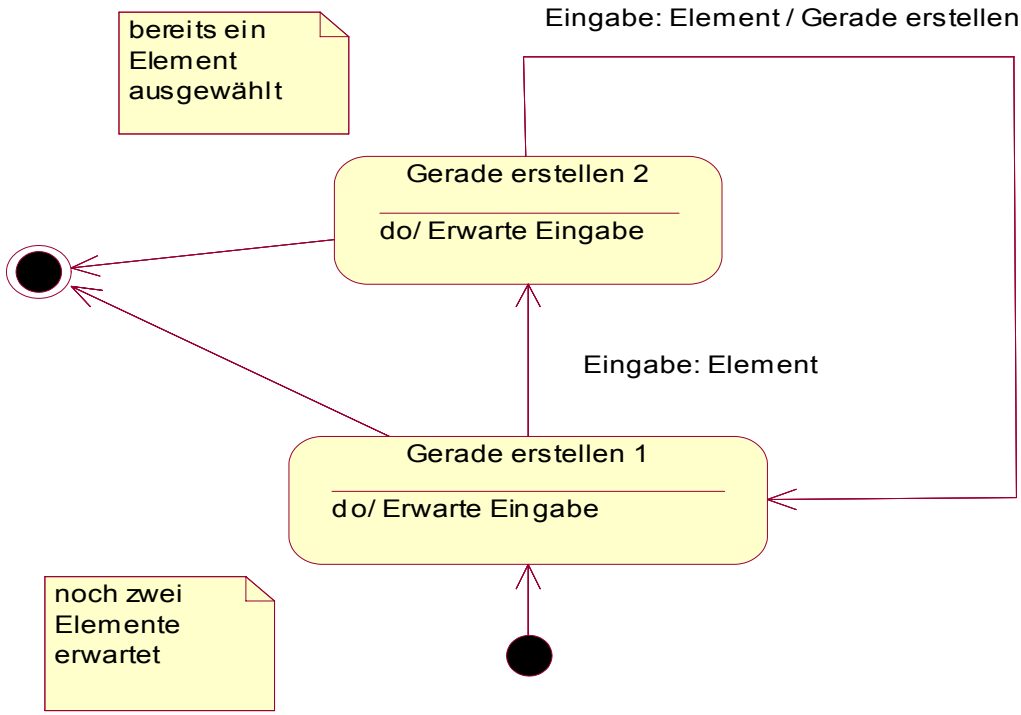

Diagramm 3

Repräsentation von Konstruktionen

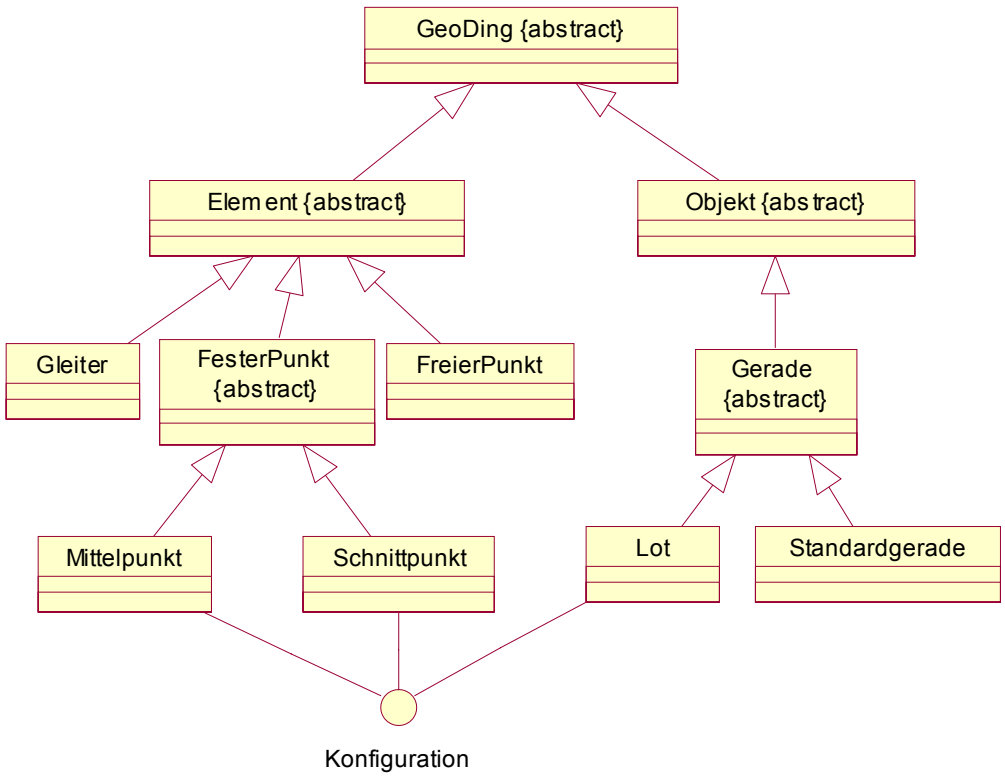

Diagramm 4

In unserem Modell unterscheiden wir zwischen Elementen – geometrisch Punkte – und Objekten, wie Geraden bzw. in späteren Programmversionen auch Kreise. Beide Begriffe beschreiben disjunkte Mengen. Es gibt jedoch eine gemeinsame (abstrakte) Oberklasse, GeoDing, in welcher in Form von abstrakten Methoden gemeinsame Grundfunktionen, z.B. Zuarbeiten für die grafische Darstellung, zur Verfügung gestellt werden können. Darüber hinaus können so beide in derselben generischen Datenstruktur untergebracht werden.

Elemente wiederum lassen sich vollständig disjunkt in Gleiter, bewegliche Punkte (beide vom Nutzer gesetzt), sowie feste Punkte (vom Programm berechnet) unterteilen. Da jeder solche feste Punkt in einer Konfiguration stehen muss, ist diese Klasse ebenfalls abstrakt. Für die erste Programmversion sind Mittelpunkte und Schnittpunkte als spezielle feste Punkte vorgesehen.

Als Objekte sind in der ersten Programmversion nur Geraden vorgesehen. Diese lassen sich wiederum vollständig disjunkt in Standardgeraden – also solche, die der Nutzer durch Auswahl zweier Elemente definiert hat – und Lote, welche durch eine andere Gerade und einen Punkt definiert sind, beschreiben.

Lote, Mittel- und Schnittpunkte implementieren das Interface Konfiguration. Darin werden Methoden versprochen, welche alle Objekte und Elemente in Konfigurationen bereitstellen

müssen. Damit können diese in generischen Datenstrukturen gemeinsam gespeichert werden, und das Vorhandensein für andere Programmteile unbedingt nötiger Funktionalität wird auch für eventuelle weitere, später implementierte Konfigurationen sichergestellt.
Dieses Modell ermöglicht also die – zumindest logisch – problemlose Integration weiterer Funktionalität. Dabei finden sich die semantischen Zusammenhänge bzw. Konfigurationen in den einzelnen Instanzen der Klassen.

So sollen alle Elemente über eine Liste von Referenzen auf alle Objekte, zu denen sie gehören oder mit denen sie in einer Konfiguration stehen, verfügen und umgekehrt. Damit ist es im Falle einer Positionsänderung eines Elementes oder Objektes möglich, an alle betroffenen anderen Elemente oder Objekte Meldungen zu senden. Falls selbige auch ihre Position ändern müssen, können ebenso weitere Nachrichten an die entsprechenden Elemente und Objekte gesendet werden, usw.

Objekte werden durch Elementmengen oder andere Objekte oder beides definiert, so z.B. eine Gerade durch zwei Elemente. Diese Elemente, durch welche ein Objekt definiert wurde, nehmen einen besonderen Stellenwert ein. Das Objekt kann nur über sie verändert werden! Z.B. können auf einer Standardgerade, die durch zwei Punkte definiert ist, dementsprechend nur noch Gleiter liegen.

Eine Konstruktion ist auf dieser Grundlage also eine Menge von GeoDingen.

MVC- Realisierung

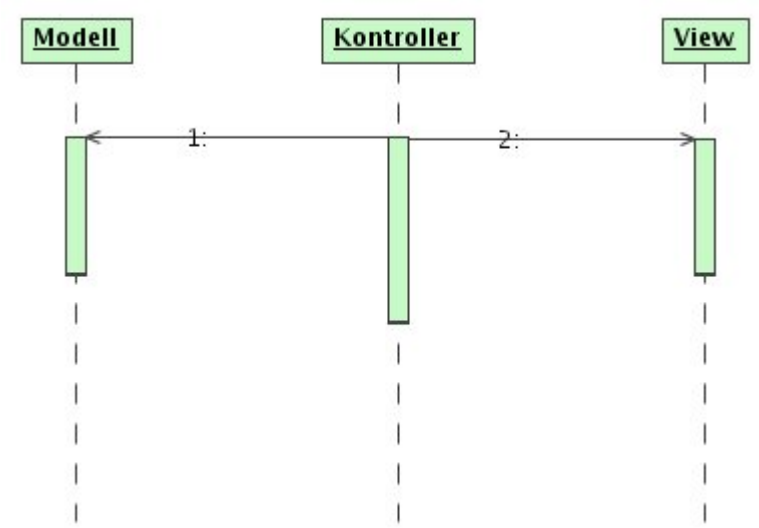

Diagramm 5

Da die Software eine Dialog-basierte Anwendung ist, müssen alle **Aktivitäten des Benutzers** abgefangen und entsprechend verarbeitet werden.
In unserem Fall müssen solche GUI-Klassen wie Buttons, Zeichenflächen und Menüs einen eigenen **MouseListener** bzw. **KeyListener** implementieren.
In der Software wird es noch eine Koordinator-Klasse **Kontroller** geben, an die alle auftretenden und in einem entsprechenden Listener filtrierten Ereignisse zusammen mit den nötigen Parametern (z.B. Koordinaten) delegiert weden. Der Kontroller besitzt das Recht, die Werte des Modells zu verändern und die Konstruktion neu zu zeichnen.
Der ganze Vorgang wird exemplarisch am **Beispiel "Punkt setzen"** erklärt:

- Der Benutzer klickt den Button "Punkt setzen" an.
- Die Klasse Button implementiert einen MouseListener, der dem Kontroller eine Botschaft zuschickt; dies heißt: "ich wurde angecklickt" :)
- Der Kontroller geht danach in den Konstruktionsmodus über. Es wird auf einen Punkt gewartet, dies bedeutet, es wird auf einen Klick auf der Zeichenfläche gewartet.
- Die Zeichenfläche verfügt auch über einen MouseListener, der abhängig von dem Anwendungszustand (Bewegung, Konstruktion, Gruppierung) einen Mouse-Klick unterschiedlich interpretieren kann. Unter den beschrieben Umständen des Beispiels werden nach einem Klick auf der Zeichenfläche die Koordinaten für einen neuen Punkt an den Kontroller geschickt.
- Nach gewisser Überprüfung (z.B. in den Koordinaten liegt schon ein Punkt, was eine Fehlermeldung auslöst,) wird ein neues Objekt Punkt erzeugt (Modell) und ein Punkt wird auf der Zeichenfläche gezeichnet (View).

4.Paket- und Klassenstruktur

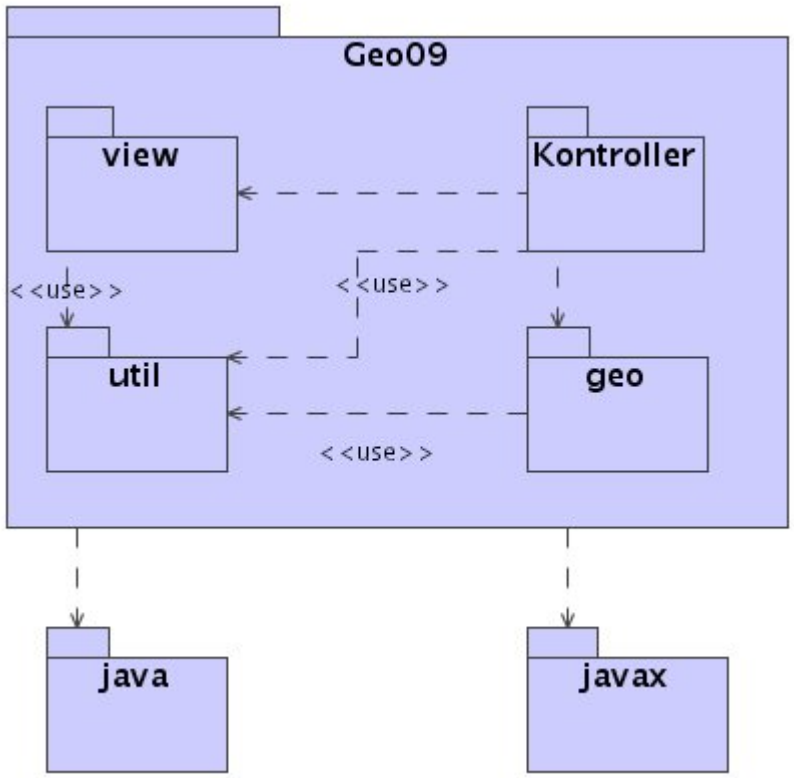

Diagramm 6

Paket	***Beschreibung***
view	Abgeleitete Swing- Klassen für GUI-Komponente und für das Zeichnen der einzelnen geo-Objekte
geo	Geometrische Elemente, Objekte und Konfigurationen

Paket	***Beschreibung***
Kontroller	Kein physisches Verzeichnis, sondern der logischen Deutlichkeit und der Wichtichkeit der Klasse Kontroller wegen wird dieser hier in die Paketstuktur reingezogen. • Konsistenzüberprüfung • Steuerung der internen Zustände
util	Klassen für mathematische Funktionen und diverse Algorithmen, meistens mit Klassenmethoden ausgestattet
java + javax	Grundlegende Java- Klassen von Sun + Swing

Bemerkung:

Wegen der Ansätze, welche das Extreme Programming-Paradigma liefert und die weitgehend im Praktikum ausgeübt werden, gehen wir in der ersten Version der Design-Beschreibung nicht genauer auf die Spezifikation der einzelne Klasse ein.
Es wird aber in dieser Version schon ziemlich genau das Modell der Software - geo-Paket - betrachtet, weil damit die Implementierung in unserer Gruppe anfängt.

Geo- Paket

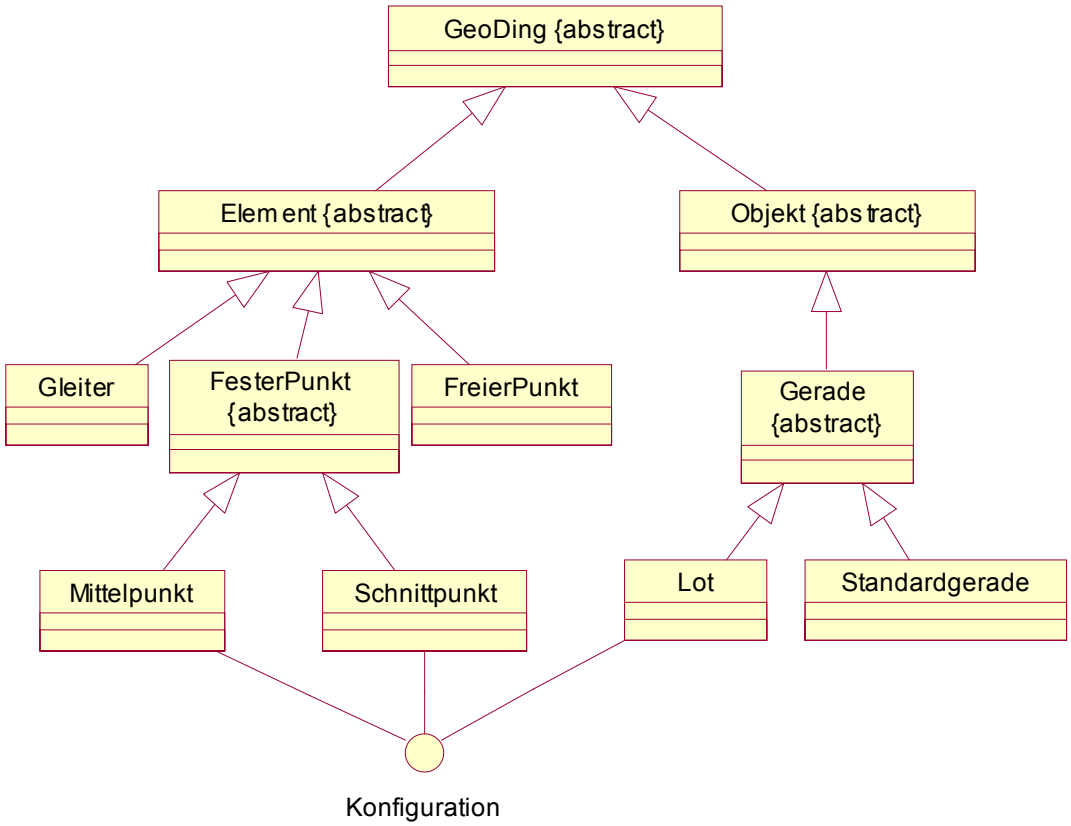

Diagramm 7

Die Beschreibung des Diagrammes findet sich im Abschnitt 3 der Design-Beschreibung (Repräsentation von Konstruktionen).