

Designbeschreibung LINEASEL Gruppe Geo04

Name	Datum	Version
Haschke, Orlamünder	04.06.2002	0.5
Schuhmann, Orlamünder	06.06.2002	1.0
Meißner	16.06.2002	2.0

<u>Inhaltsverzeichnis</u>	<u>1</u>
<u>1. Allgemeines</u>	
<u>1.1 Kurzcharakterisierung</u>	<u>2</u>
<u>1.2 Systemvoraussetzungen</u>	<u>2</u>
<u>1.3 Abgrenzungen</u>	<u>2</u>
<u>2. Produktübersicht</u>	<u>2</u>
<u>3. Grundsätzliche Design-Entscheidungen</u>	<u>2</u>
<u>3.1 Allgemeines</u>	<u>2</u>
<u>3.2 Geometrische Elemente (Modellsicht)</u>	<u>3</u>
<u>3.2.1 Visualisierung von geometrischen Objekten (Darstellung)</u>	<u>3</u>
<u>3.3 Initialisierungsphase</u>	<u>3</u>
<u>3.4 Konstruktionsmodus/ Zugmodus</u>	<u>4</u>
<u>3.5 Projekt laden</u>	<u>4</u>
<u>3.6 Projekt speichern</u>	<u>4</u>
<u>3.7 Programmablaufplan</u>	<u>5</u>
<u>4. Paket und Klassenstruktur</u>	<u>6</u>
<u>4.1 Das Paket model</u>	<u>6</u>
<u>4.2 Das Paket view</u>	<u>7</u>
<u>4.3 Das Paket controller</u>	<u>8</u>

1. Allgemeines

1.1 Kurzcharakterisierung

LINEASEL ist eine vorwiegend mausgesteuerte grafische Java-Applikation mit der sich geometrische Konfigurationen visualisieren lassen. Die geometrischen Konfigurationen lassen sich mit der Maus in Zeichenfenstern erstellen.

1.2 Systemvoraussetzungen

LINEASEL ist durch die Implementierung in Java plattformunabhängig.
Es wird das JRE 1.4 benötigt.

1.3 Abgrenzungen

In der ersten Version sind noch keine Kreise vorgesehen. Ebenso sind Berechnungen wie „Lot fällen“ oder „Winkel berechnen“ nicht geplant.

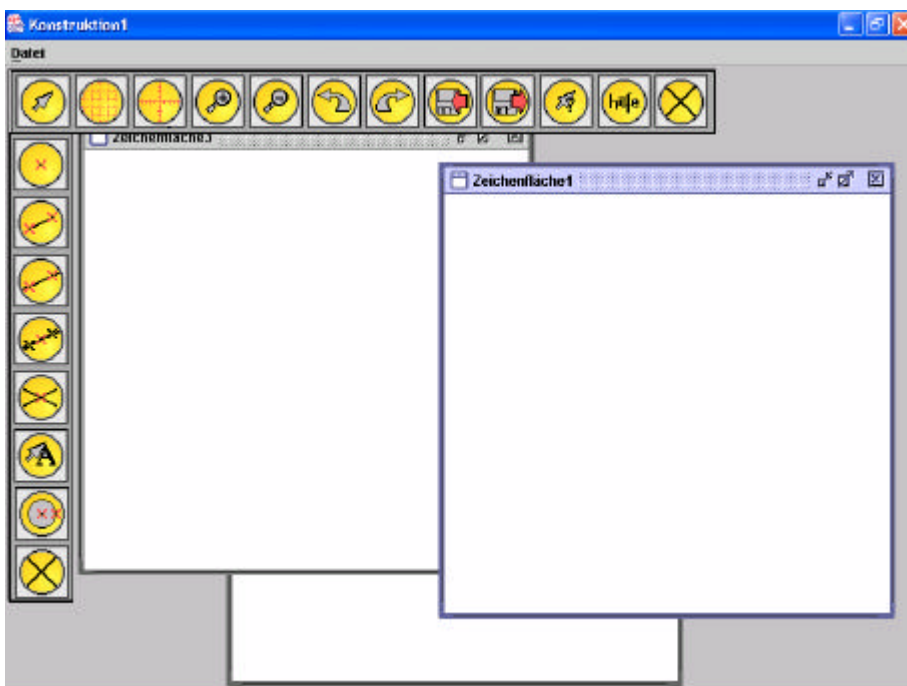
2. Produktübersicht

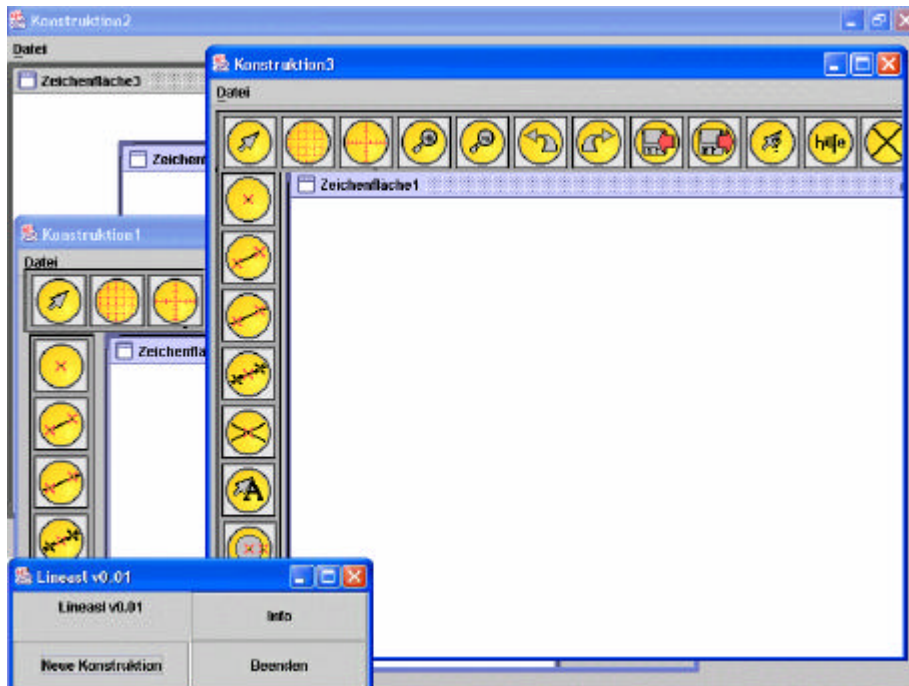
Während der Initialisierung wird das Startfenster aufgebaut.



Das Startfenster bietet die Möglichkeiten, eine neue Konstruktion anzulegen, eine Aboutbox zu öffnen, bzw. das Programm wieder zu schließen.

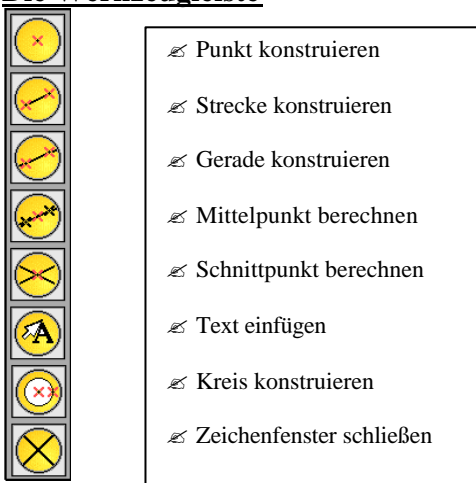
Wird eine neue Konstruktion angelegt, so wird ein Konstruktionsfenster mit einem leeren Zeichenfenster geöffnet. Nun besteht die Möglichkeit entweder eine bestehende Konstruktion über einen „Datei öffnen“ – Button (im Konstruktionsfenster) zu laden oder eine per Maus selbst zu konstruieren.





Diese Konstruktionen können durch Anklicken und Ziehen von Punkten dynamisch verändert werden. Über einen „neues Zeichenfenster“ – Button können weitere Zeichenfenster geöffnet werden, die verschiedene Sichten auf die selbe Konstruktion ermöglichen. Zwischen den Zeichenfenstern kann per Maus hin und her geschaltet werden.

Die Werkzeugleiste



Die Menuleiste



von links nach rechts:

- ☞ Pfeil zum verschieben von Objekten
- ☞ Koordinatenraster ein/ausblenden
- ☞ Hauptachsen ein/ausblenden
- ☞ Hineinzoomen, skalieren
- ☞ Herauszoomen, skalieren
- ☞ Schritt zurück, undo
- ☞ Schritt vor, redo
- ☞ Konstruktion speichern
- ☞ Konstruktion laden
- ☞ Hilfe zu Thema
- ☞ Hilfe
- ☞ Konstruktion schließen

3. Grundsätzliche Design-Entscheidungen

3.1 Allgemeines

Der Paket- und Klassenstruktur liegt die Idee der Trennung von Daten - Ansicht - Kontrolleinheit (MVC) zugrunde.

Die graphische Oberfläche wird mit Java Swing Klassen, Fenster-Technik und dem Ereignis-Aktions-Konzept realisiert. Die Realisierung der Zeichenfläche folgt nicht durch Java standardisierte Überlegungen, für andere einfache Grafikelemente wird auf Swing-Standards zurückgegriffen.

Ein Parser und Übersetzer sind an der Stelle notwendig, wo vorhandene Konfigurationen geladen bzw. gespeichert werden sollen, da dazu Dateiströme (linear geschriebene Daten) in die interne objektorientierte Struktur umgewandelt werden müssen bzw. die genau entgegengesetzte Richtung (objektorientierte Struktur in Datei schreiben) abgesichert sein muss.

Außerdem sollten auch Klassen zum Anzeigen von Dialogen und Meldungen (Fragen, Informationen, Fehler) vom eigentlichen Inhalt des Programms getrennt werden, da eine solche saubere Trennung mehrere Vorteile wie Weiterverwendung von Code in anderen Programmen und mehr Möglichkeiten zur Erweiterung des Codes bietet.

3.2. Geometrische Elemente (Modellsicht)

In unseren Augen ist es die beste Strategie, was die Speicherung der Daten betrifft, wenn die geometrischen Elemente direkt aufeinander aufbauen. Dies ist günstig, da eine Gerade bzw. Strecke durch zwei Punkte, ein Polygon durch n Punkte definiert ist.

Dazu entwickelt man zunächst eine Klasse GeoElement, welche als Attribut eine Instanz der Klasse GeoGruppierung enthält. Sie „verknüpft“ das Element (über eine Koordinate mit einem anderen). Die Klasse GeoGruppierung wiederum enthält die Attribute GeoPunkt (über diese Koordinate wird verknüpft), GeoElement (dahin wird verknüpft), GeoGruppierung (eventuell nächste Verknüpfung). Unterklassen von GeoElement sind GeoPunkt, GeoStrecke, GeoGerade, (evtl. GeoKreis) und GeoPolygon. Durch die Klasse GeoPolygon sind automatisch alle geometrischen Figuren wie Dreiecke, Viereck und spezielle und gemeine Vielecke abgedeckt.

Die Speicherung der Parameter funktioniert folgendermaßen: In GeoPunkt werden die Koordinaten des Punktes gespeichert und in GeoStrecke, GeoGerade sowie GeoPolygon werden Objekte von GeoPunkt instanziiert. Dadurch können verschiedene Objekte auch wieder zu einem Verbund werden.

Diese Geometrischen Objekte liegen im Paket model.

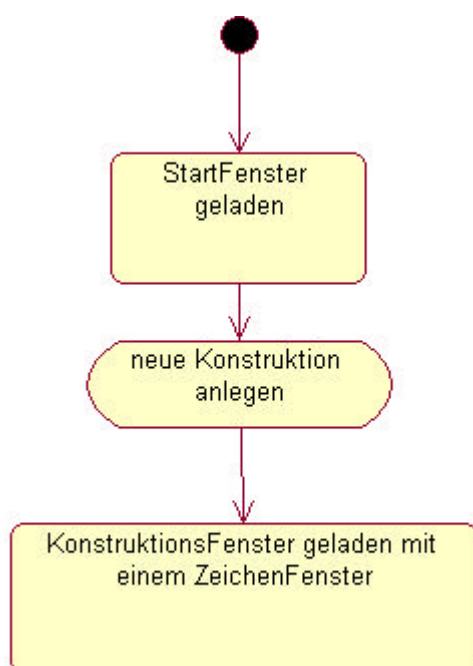
3.2.1 Visualisierung von geometrischen Objekten (Darstellung)

Das Zeichnen bzw. die Visualisierung der vorhandenen geometrischen Objekte erfolgt durch Klassen in einem von der Datenhaltung (Paket model) getrennten Paket view. In den Klassen des Paketes finden auch Umrechnung von Konfigurationskoordinaten in aktuell anzuzeigende Fensterkoordinaten und umgedreht statt.

Durch diese Trennung zwischen model und view können die Klassen von model auch zur objektgebundenen Speicherung geometrischer Daten in anderen Programmen genutzt werden.

3.3 Initialisierungsphase

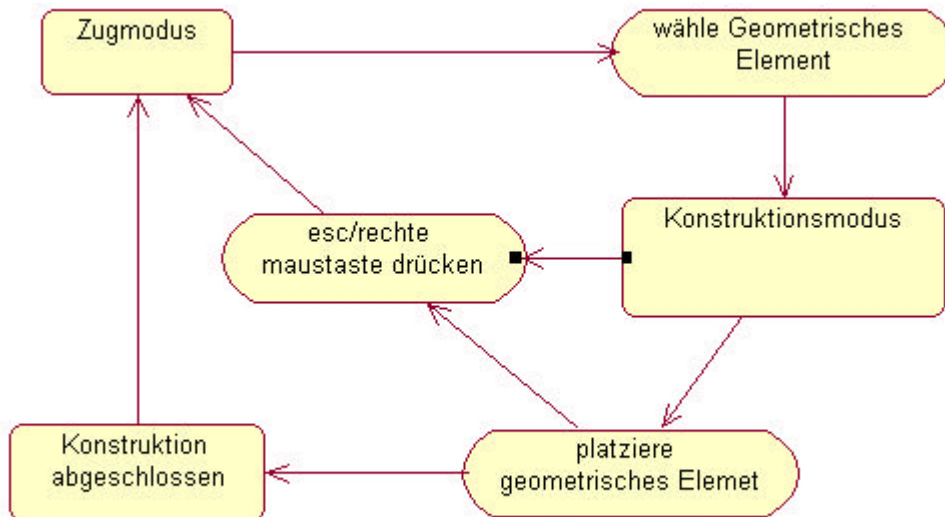
Nach Start von LINEASEL wird lediglich ein Hauptkontroll-Panel der Klasse StartFenster initialisiert, von dem der Nutzer die Möglichkeiten hat, neue Konstruktionen (Konfigurationen) zu starten, sich eine Programm-Info anzeigen zu lassen und das Programm zu beenden.



Wählt der Benutzer „neue Konstruktion“ im StartFenster, so wird ein KonstruktionsFenster mit einem ersten ZeichenFenster geladen. In diesem Fenster wird auch die leere Konfiguration (also nichts) angezeigt.

3.4 Konstruktionsmodus/ Zugmodus

Im Konstruktionsmodus wird ein geometrisches Element - welches gezeichnet werden soll - angewählt und platziert. Im Zugmodus wird dieses durch Ziehen an (Eck)punkten verändert. Startzustand ist standardmäßig der Zugmodus.



3.5 Projekt laden

Es wird der Parser initialisiert, welcher die vorhandene Datei auf Fehler prüft und in Objekte der Konfiguration umformt. Es wird das Projekt mit Projektfenster initialisiert und darin ein erstes Zeichenfenster geladen. In diesem Fenster wird in seinem durch Standardeinstellung gegebenen Bereich die vorhandene Konfiguration angezeigt. Dazu müssen Klassen aus view initialisiert werden.

3.6 Projekt speichern

Hier setzt der Parser den Objektcode in einen Dateistrom (linear geschriebene Daten) um. Diese werden in Textform persistent in Dateien gespeichert.

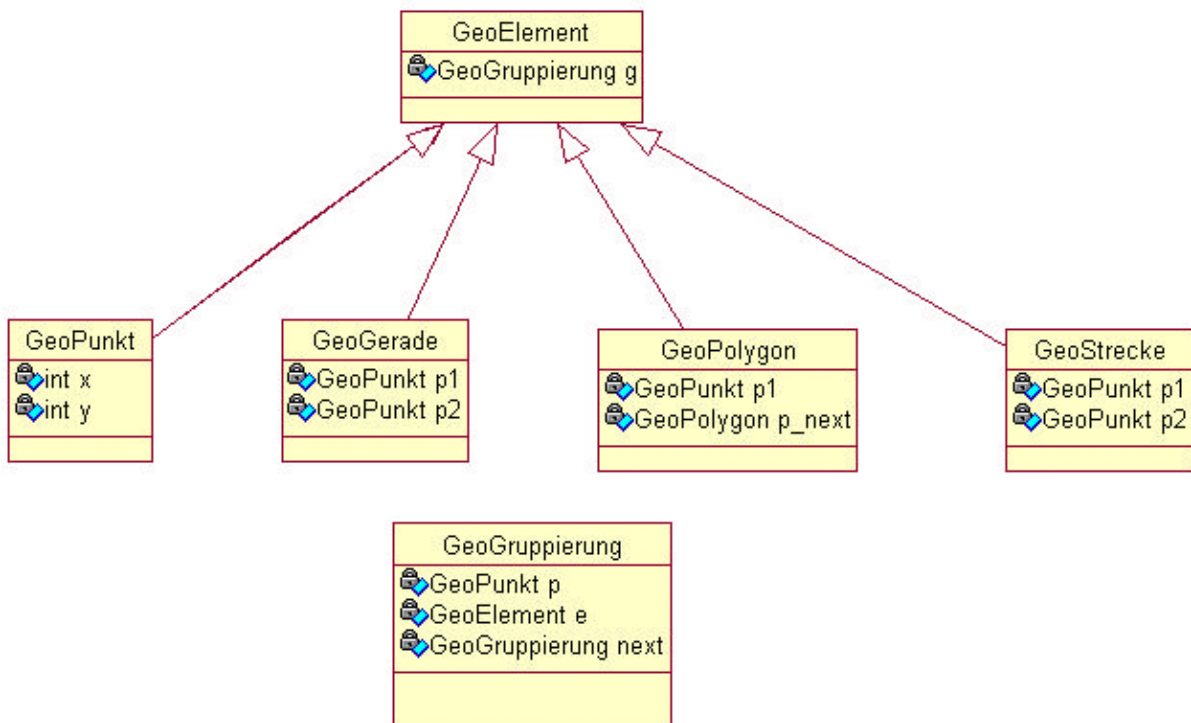
4. Paket und Klassenstruktur

Es besteht zwecks MVC-Trennung aus den Paketen model, view, controller, und eine Hauptklasse in der sich die main() Methode befindet.

Diese Hauptklasse ist für die Initialisierung des Programms zuständig.

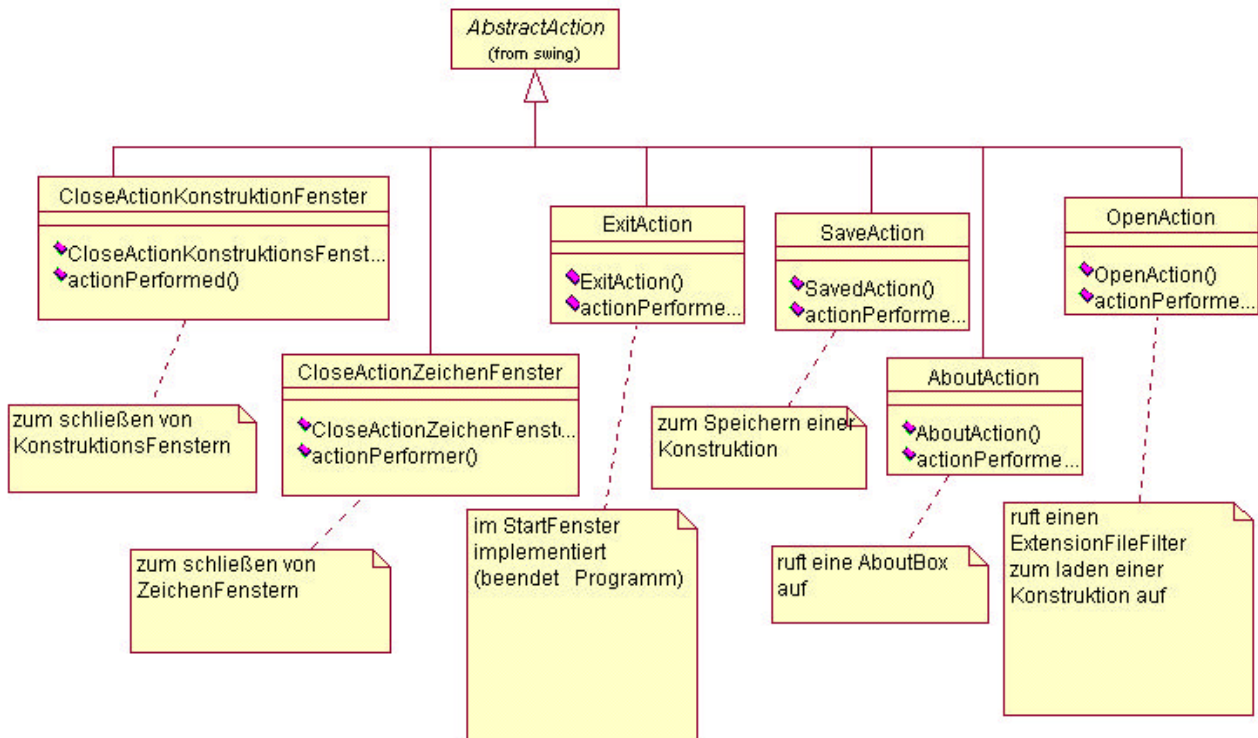
4.1 Das Paket model

Im Paket model sind die Klassen der geometrischen Elemente zusammengefasst. Die Klassen der geometrischen Elemente enthalten die Attribute die für ihre eindeutige Lage im Weltkoordinatensystem notwendig sind, z.B. GeoPunkt zwei Punkte x;y, oder GeoStrecke mit Punkt;Punkt. Zusätzliches Attribut ist ein Objekt der Klasse Gruppierung welches zwei oder mehr geometrische Elemente punktweise bei gleichen Koordinaten zusammenfasst .

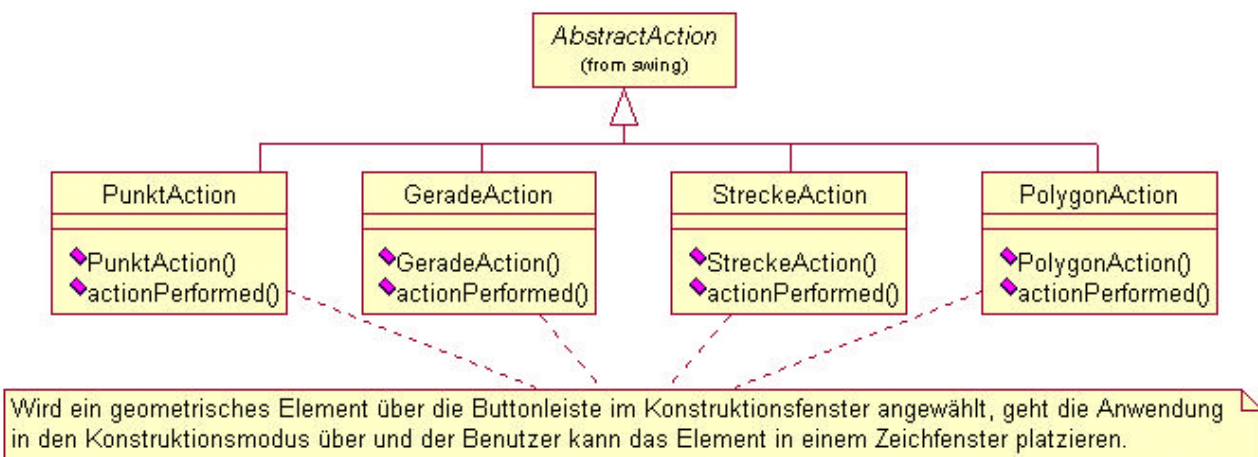


4.2 Das Paket view

Diese Paket enthält alle Fenster- und Actionklassen sowie die Klassen, die die Visualisierung geometrischer Objekte steuern. Zu den Fensterklassen gehören StartFenster, KonstruktionsFenster sowie ZeichenFenster. In diesen Fenstern werden Aktionen abgehört. Nach dem Event-Action-Konzept gibt es zu jeder Aktion eigene Aktionsklasse, welche die Swingklasse AbstractAction erweitert.

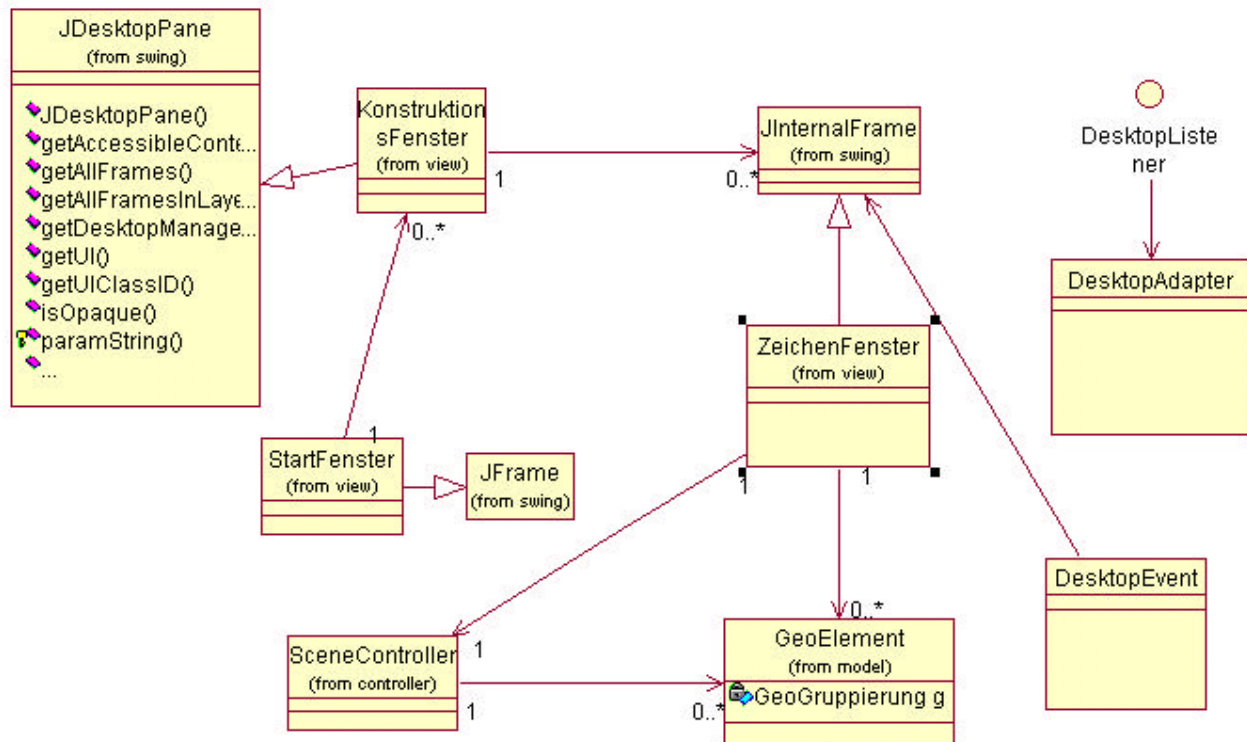


Außerdem müssen die Buttons zum Auswählen eines geometrischen Elements abgehört werden:



Beim Start des Programms wird die Klasse StartFenster – abgeleitet von JFrame - instanziiert und damit das Hauptfenster der Applikation angelegt.

Über dieses Fenster kann ein Konstruktionsfenster aufgerufen werden, welches eine Instanz der Klasse JDesktopPane ist. In diesem Konstruktionsfenster können beliebig viele Zeichenfenster - abgeleitet von JInternalFrame – geöffnet werden.



4.3 Das Paket controller

Im controller-Paket befinden sich die Klassen zur Berechnung. Dazu gehören:

- die Umrechnung von Fenster- in Weltkoordinaten und zurück
- die Berechnung aller Koordinaten der platzierten geometrischen Elemente und deren Vergleich mit den Koordinaten der Mausposition (umgerechnet in Weltkoordinaten)
- die interne Gruppierung geometrischer Elemente (falls beim Platzieren im Konstruktionsmodus die Mauskoordinaten mit den Koordinaten eines schon bestehenden geometrischen Elements zusammenfallen)
- die Neuberechnung aller Punkte der geometrischen Elemente (in Weltkoordinaten) nach einer Veränderung eines Punktes im Zugmodus