

Designbeschreibung

Softwaretechnik-Praktikum SS 2003

Gruppe: Geo01

Version 1.2

	Autor	Datum	Status	Kommentar
1.0	Kretschmann	07.07.2003	Draft	Nachbearbeitung 3.1.4
1.1	Reusche	22.06.2003	Final	Klassendiagramme, Layout

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



1. Allgemeines

1.1 Kurzcharakterisierung

Das von uns erstellte Programm ist eine menügesteuerte graphische JAVA – Applikation, mit der sich geometrische Konstruktionen darstellen lassen. Diese Dynamische Geometrie Software soll zur Veranschaulichung und Lösung geometrischer Probleme eingesetzt werden.

1.2 Systemvoraussetzungen

Benötigt wird das Java SDK 1.4. Ansonsten sind keine besonderen Anforderungen an Hard- oder Software gestellt.

1.3 Beschreibung der Produktumgebung

Um das Programm ausführen zu können wird ein Betriebssystem benötigt, welches ein Java Umgebung besitzt. Es ist nicht genau spezifiziert, wo dieser Arbeitsplatz steht, also ob es sich um einen PC oder eine Workstation handelt, oder ob das ganze in einem Arbeitszimmer einer Privatperson oder in einer großen Firma steht, ist nicht vorgeschrieben.

1.4. Abgrenzung

Das Produkt ist eine Einzelplatzanwendung und muss nicht netzwerkfähig sein.

2. Produktübersicht

Die Dynamische Geometriesoftware ist primär dazu in der Lage, eine neue Konstruktion anzulegen, indem ein erstes geometrisches Objekt platziert wird. In diese Konstruktion können weitere geometrische Objekte eingefügt werden. Jede Konstruktion erfolgt auf einer eigenen Zeichenoberfläche.

Die Platzierung eines geometrischen Objekts erfolgt durch die Maus. Beim Anlegen eines Punktes wird dieser der Konstruktion hinzugefügt und an der entsprechenden Koordinate auf der Zeichenoberfläche dargestellt. Es ist auch möglich, die Koordinaten eines Punktes im Dialog per Tastatur einzugeben. Weitere geometrische Objekte, im Konkreten die Strecke, Gerade und Halbgerade, müssen mit mindestens zwei Punkten aus der Konstruktion in Zusammenhang stehen. Diese Punkte werden entweder neu angelegt, oder es werden bereits bestehende Punkte verwendet. Nach jedem Konstruktionsschritt wird das jeweilige neue geometrische Objekt der Konstruktion hinzugefügt und damit auf der Zeichenoberfläche dargestellt.

Es ist möglich, bereits bestehende Punkte zu löschen. Ebenso können Punkte verschoben werden.

Generell werden Veränderungen in der Konstruktion nach jedem Bewegen oder Löschen eines Punktes auf der Zeichenoberfläche neu gezeichnet. Zur Unterscheidung der geometrischen Objekte bei der Erzeugung kennt das Programm verschiedene Konstruktionsmodi. Je nachdem, welcher Konstruktionsmodus gewählt wird, kann ein bestimmtes geometrisches Objekt der Konstruktion hinzugefügt werden.

Das Programm befindet sich während der Laufzeit immer entweder in einem der Konstruktionsmodi, im Berechnungsmodus oder im Bearbeitungsmodus. Der Benutzer kann zwischen den einzelnen Modi mit Hilfe der graphischen Oberfläche in Form von Schaltflächen, die neben der Zeichenoberfläche oder in einem Menü zu finden sind, umschalten. Ein Modus bleibt so lange aktiv, bis ein anderer ausgewählt wird.

Um dem Benutzer die Arbeit mit dem Programm zu erleichtern, können sowohl die Koordinatenachsen des Koordinatensystems als auch ein Gitternetz vom Benutzer beliebig ein- und ausgeblendet werden. Beide Hilfsmittel beziehen sich auf dasselbe Koordinatensystem, das auch zur Bestimmung der Koordinaten der Punkte verwendet wird.

Der Benutzer wird zu keiner Zeit in der Lage sein, ein geometrisches Objekt direkt auf der Zeichenoberfläche per Maus markieren zu können. Die Software wird die Möglichkeit bieten, einen Punkt aus einer Liste aller zur Konstruktion gehörenden Punkte auszuwählen. Diese Objektreferenzliste wird automatisch beim Öffnen der Zeichenoberfläche erstellt. Es wird jeder Punkt, der während eines Konstruktionsschrittes angelegt wird, der Liste hinzugefügt.

Um auch größere Konstruktionen übersichtlich gestalten zu können, ist es möglich, die Beschriftung eines

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



Punktes nach Wunsch ein- bzw. auszublenden. Außerdem kann jedem geometrischen Objekt eine eigene Farbe zugewiesen werden, in welcher das Objekt dann auf der Zeichenoberfläche dargestellt wird, so dass der Anwender eine bessere Übersicht erhält.

Um den schnellen und problemlosen Einstieg in die Benutzung des Produktes zu ermöglichen, steht eine vom Programm aufrufbare Hilfe zur Verfügung. Diese sollte alle etwaig auftretenden Fragen beantworten können.

3. Grundsätzliche Design Entscheidungen

Wir haben unseren Programmcode nach dem MVC Konzept unterteilt, d.h. der Programmcode wurde einer Dreiteilung in die Klassen Modell, Controller und View unterzogen. In dem Programmcode befinden sich also drei verschiedene Arten von Klassen:

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

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

Die Anwendung des MVC Konzeptes haben wir gewählt um modular programmieren zu können. Wobei es uns hauptsächlich darum ging, dass sich das MVC leicht umsetzen lässt, wenn man mit Hilfe des Pair-Programming arbeitet. Die einzelnen Gruppen arbeiten unabhängig voneinander, d.h. sie müssen keine Rücksicht auf den Fortschritt einer anderen Gruppe nehmen.

Das Grundprinzip der Datenverarbeitung im Programm, also Grundprinzip der Konstruktion und aller geometrischen Objekte wurde folgendermaßen gewählt und festgelegt:

Die Basisklasse, die die Grundlage der gesamten Datenspeicherung bildet, ist die Klasse geometrisches Objekt. Jedes geometrische Objekt (Punkt, Gerade, usw.) ist eine Instanz der Klasse geometrischesObjekt oder eine Instanz der Unterklassen der Klasse geometrischesObjekt. Allerdings bildet die Grundlage für die Berechnung der Positionen der geometrischen Objekte die Klasse Punkt, denn diese Klasse ist die einzige in der konkret Zahlenwerte gespeichert sind, die verändert werden können.

Diese Tatsache ist darauf begründet, dass jedes geometrische Objekt mit Hilfe von Punkten eindeutig beschrieben werden kann. Dies sind die double Werte x und y des Punktes. Alle anderen Arten von geometrischen Objekten berufen sich bei der Abfrage ihrer bestimmenden Daten immer auf einen oder mehrere mit diesem Objekt verbundenen Punkte. Das heißt zum Beispiel, dass wenn man auf die Daten einer Gerade zugreifen möchte, man auf die beiden Punkte, die die Gerade bilden, zugreifen muss, und deren x und y Werte ausliest. Ebenso ist es bei etwas komplexeren Objekten, wie einem Mittelpunkt. Dieser hat keine konkreten Zahlenwerte gespeichert, sondern in den Methoden getX und getY, die die konkreten Koordinaten, bei denen sich der Mittelpunkt im Augenblick befindet, zurückgeben, ist die Berechnungsvorschrift abgespeichert, mittels derer die Koordinaten ermittelt werden. Diese Berechnungsvorschriften wiederum beziehen sich auf die fest gespeicherten Koordinaten der Punkte, zwischen denen der Mittelpunkt definiert ist.

Zusammenfassend kann man also sagen, dass das Grundprinzip der hier verwendeten dynamischen Verwaltung der geometrischen Objekte, die Neuberechnung der geometrischen Objekte ist, die keine Punkte sind. Dieses Konzept lässt sich durchaus auch auf komplexere Konstruktionen, wie das Anlegen einer Parallelen zu einer Gerade durch einen Punkt, übertragen. Wie dies in späteren Versionen des Programms erfolgen würde, ist im UML-Diagramm zum Programm Geo01 zu sehen, wo der Entwurf, farblich getrennt vom Rest des Programms, zu sehen ist.

Natürlich entsteht dadurch eine etwas geringere Effizienz in der Verarbeitungsgeschwindigkeit des Programms, doch wenn man die Leistung heutiger Computersysteme berücksichtigt, so ist dieser Nachteil kaum von Bedeutung. Zudem lassen sich große Datenbestände mit dieser Lösung der dyna-

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



mischen Verwaltung bedeutend besser bearbeiten und verwalten. Bei sehr großen Konstruktionen (100 geometrische Objekte und mehr) ist es wohl schon aus Gründen der reinen Übersichtlichkeit, sinnvoller das Anlegen und Verändern von geometrischen Objekten nicht mehr per direkter Eingabe (sei es per Maus oder Tastatur), sondern über eine Skriptsprache zu bewerkstelligen. Und bei solch großen Datenmengen und einer derartigen Eingabe ist das hier verwendete Konzept eindeutig überlegen. Denn es müssten nur einzelne Punkte verschoben werden um komplexe Konstruktionsänderungen durchzuführen. Weitere Berechnungen sind nicht erforderlich! Lediglich, am Ende der Bearbeitung muss ein einziges mal die Berechnung aller geometrischen Objekte erfolgen. Für spätere Programmversionen ist solch eine Art der Programmanwendung durchaus in Erwägung zu ziehen. Zudem hat man durch diese Art der Datenspeicherung und Erzeugung der aktuell benötigten Daten den Vorteil, dass ein Großteil des Rechenaufwandes erst beim Zeichnen nötig ist.

Grundsätzliche Funktionsweise des Programms

Zu Beginn des Programms, also nach Starten der Klasse Geo01, wird eine Instanz der Klasse HauptController angelegt, mit der das Hauptfenster des Programms aufgebaut wird und alle Schaltelemente des Programms angelegt werden. Die Ereignisverarbeitung, für die im HauptController angelegten Schaltelemente, übernimmt die Klasse HauptController selbst.

3.1.

Wenn der Button (oder der Menüeintrag) "Neue Konstruktion" angeklickt wird, reagiert die Ereignisverarbeitung des HauptControllers folgendermaßen:

Eine neue Instanz der Klasse Konstruktion wird angelegt, und ebenso je eine Instanz der Klassen KonstruktionsView, KonstruktionsController, PunkteReferenzView und PunkteReferenzController.

Es wurde also eine neue Konstruktion erstellt, mit der nun gearbeitet werden kann.

Im Einzelnen bewirken die eben angelegten Instanzen der Klassen folgendes:

- Die Instanz der Klasse Konstruktion ist der Ort der Datenspeicherung der geometrischen Objekte der Konstruktion, es werden also Speicherstrukturen für die geometrischen Objekte instanziiert. Dies geschieht mittels einer Liste, umgesetzt mit Hilfe der Java-Klasse Vektor, in der jedes einzelne existierende geometrische Objekt referenziert werden soll. Zu Beginn ist die Liste leer.
- Die Instanz der Klasse KonstruktionsView dient der graphischen Darstellung der Konstruktion in einem JInternalFrame, das im Programm-Hauptfenster angelegt wird. Dies dient dem Programm als Zeichenoberfläche.
- Ebenso wird die Instanz der Klasse PunkteReferenzView angelegt, welche zur Darstellung der geometrischen Objekte in einer JListe in einem JInternalFrame des Hauptfensters, dient.
- Die Instanz der Klasse KonstruktionsController dient später zur Ereignisverarbeitung der Ereignisse, die in Verbindung mit der Klasse KonstruktionsView auftreten.
- Die Instanz der Klasse PunkteReferenzController dient später zur Ereignisverarbeitung der Ereignisse, die in Verbindung mit der Klasse PunkteReferenzView auftreten.

Nun stehen, nach Anlegen der Konstruktion, folgende Aktionsmöglichkeiten in beliebiger Kombination zur Auswahl. Die Aktionsmöglichkeiten entsprechen jeweils den einzelnen Programmmodi, wie sie auch im Zustandsdiagramm zu sehen sind. Es ist jeweils noch aufgeführt, was dabei intern im Programm alles geschieht, und welche Klassen wie beteiligt sind, also wie die Klassen miteinander interagieren.

3.1.1 „Punkt anlegen“

Man muss im Programmmodus Punkt-Anlegen sein um dies durchführen zu können.

Entweder wird in das Konstruktionsfenster geklickt, und somit die Koordinaten des neuen Punktes vom KonstruktionsController ermittelt, der ja das Ereignis, das beim Klicken auftritt verarbeitet, oder die Koordinaten des Punktes werden direkt als Zahlen, im dafür vorgesehenen Eingabefeld im Hauptfenster, eingegeben. In beiden Fällen verfügt einer der Controller dann über die Koordinaten des anzulegenden Punktes. Im Folgenden ruft dieser Controller nun die punktAnlegen-Methode der Konstruktion auf und übergibt die

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



eingelassenen Koordinaten. Nun wird in der Konstruktion eine neue Instanz der Klasse Punkt angelegt, und diese der Liste der geometrischen Objekte hinzugefügt. Beim Anlegen des neuen geometrischen Objektes, wird von der Konstruktion die Methode konstruktionsAenderung aufgerufen, welche allen in der View-Liste eingetragenen View-Instanzen mitteilt, dass sich etwas an den geometrischen Objekten, in der Konstruktion, geändert hat. Daraufhin stellen alle Views den sichtbaren Bereich der Konstruktion mit den neuen Daten dar. Dieses Aktualisierungsprinzip erfolgt bei jeder Änderung die an der Konstruktion vorgenommen wird, deshalb wird dies hier nicht jedes Mal wieder erneut erwähnt.

3.1.2 „Mittelpunkt anlegen“

Man muss im Programmmodus Mittelpunkt-Anlegen sein um dies durchführen zu können.

Das Anlegen eines Mittelpunktes erfolgt ähnlich wie das Anlegen eines Punktes nach 4.1.1, mit dem Unterschied, dass zwei Punktkoordinaten angegeben werden müssen, oder dass zwei bisher existierende Punkte ausgewählt werden müssen. Dies muss über die Objektreferenzliste erfolgen, die jedes Ereignis an den PunkteReferenzController übergibt, der dann die beiden Punkte ermittelt und Referenzen auf diese als Parameter der Methode mittelPunktAnlegen der Konstruktion übergibt. Diese legt eine Instanz der Klasse Mittelpunkt an. Die Referenzen auf die Punkte, zwischen denen der Mittelpunkt angelegt werden soll, werden in die beiden entsprechenden Attribute des Mittelpunktes gespeichert.

Zu erwähnen wäre hier noch das Prinzip der Mittelpunktdarstellung. Die zur Darstellung nötigen Koordinaten des Mittelpunktes sind, im Gegensatz zu Punkten, nicht in der Instanz gespeichert, da sich diese in Abhängigkeit der Koordinaten der Ursprungspunkte des Mittelpunktes, ständig ändern. Also gibt es, anstelle der Attribute, die beiden Methoden getX und getY, die die aktuellen Mittelpunktkoordinaten berechnen, und sie an die Stellen des Aufrufes der Methoden zurückgeben. Mittelpunkte erscheinen ebenso wie Punkte in der PunktReferenzListe, und man kann sie genau so verwenden wie Punkte, auch wenn im Folgenden nur von Punkten die Rede ist, so ist dies als eine Art Oberbegriff für Punkte und Mittelpunkte aufzufassen.

3.1.3 „Schnittpunkt anlegen“

Man muss im Programmmodus Schnittpunkt-Anlegen sein um dies durchführen zu können.

Das Anlegen eines Schnittpunktes erfolgt wie das Anlegen eines Mittelpunktes, mit dem Unterschied, dass anstelle von zwei Punkten zwei Geraden/Halbgeraden/Strecken ausgewählt werden. Das Auswählen dieser geometrischen Objekte erfolgt über die Objektreferenzliste. Das Prinzip der Darstellung ist das gleiche wie bei Mittelpunkten, nur dass hier als Berechnungsgrundlage die Punkte dienen, die die Geraden/Halbgerade/Strecken aufspannen. Schnittpunkte erscheinen ebenso wie Punkte in der PunktReferenzListe, und man kann sie genau so verwenden wie Punkte, auch wenn im Folgenden nur von Punkten die Rede ist, so ist dies als eine Art Oberbegriff für Punkte und Mittelpunkte aufzufassen.

3.1.4 „Gleiter anlegen“

Man muss im Programmmodus Gleiter-Anlegen sein um dies durchführen zu können.

Das Anlegen eines Gleiters erfolgt wie das Anlegen eines Schnittpunktes, nur mit dem Unterschied dass hier nur eine Gerade aus der Objektreferenzliste ausgewählt werden muss. Allerdings ist des Weiteren noch eine Koordinatenangabe (sei es per direkter Angabe der Koordinaten oder durch klicken in der Zeichenfläche) erforderlich, aus der der Darstellungsfaktor des Gleiters berechnet wird. Dieser Faktor ist zur Berechnung der x- und y-Werte des Gleiters direkt erforderlich. Die Ermittlung der x- und y-Werte erfolgt ähnlich wie bei Mittel- und Schnittpunkten.

Gleiter erscheinen ebenso wie Punkte in der PunktReferenzListe, und man kann sie genau so verwenden wie Punkte, auch wenn im Folgenden nur von Punkten die Rede ist, so ist dies als eine Art Oberbegriff für Punkte und Mittelpunkte aufzufassen.

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



3.1.5 „Gerade anlegen“

Man muss im Programmmodus Gerade-Anlegen sein um dies durchführen zu können.

Der Benutzer wählt auf eine der Beiden bekannten Arten zwei existierende Punkte aus. Die entsprechende Auswertung der Ereignisse wird durch den PunkteReferenzController durchgeführt. Somit werden auf die eine oder andere Art zwei Punktkoordinaten ermittelt, die als Parameter der Methode geradeAnlegen, welche dann ein geometrisches Objekt der Klasse Gerade anlegt, übergeben werden. Neu erstellte Punkte werden mit punktAnlegen angelegt und bereits vorhandene Punkte werden in der Gerade referenziert. Die neue Gerade wird in die Liste der geometrischen Objekte aufgenommen.

3.1.6 „Halbgerade anlegen“

Man muss im Programmmodus Halbgerade-Anlegen sein um dies durchführen zu können.

Es geschieht fast das gleiche wie beim Anlegen der Geraden, nur das die Methode halbgeradeAnlegen anstelle der Methode geradeAnlegen aufgerufen wird, welche veranlasst, dass das neue Objekt eine Instanz der Klasse Halbgerade ist, und damit auch der erste der gewählten oder angelegten Punkte in dem Attribut begrenzungs punkt_1 referenziert wird.

3.1.7 „Strecke anlegen“

Man muss im Programmmodus Strecke-Anlegen sein um dies durchführen zu können.

Es geschieht fast das gleiche wie beim anlegend er Halbgeraden, nur das die Methode streckeAnlegen anstelle der Methode halbgeradeAnlegen, aufgerufen wird, welche veranlasst das das neue Objekt eine Instanz der Klasse Strecke ist, und damit auch der zweite der gewählten oder angelegten Punkte in dem Attribut begrenzungs punkt_2 referenziert wird.

3.1.8 „Punkt löschen“

Man muss im Programmmodus Punkt-Löschen sein um dies durchführen zu können.

Der Benutzer wählt einen Punkt aus der Objektreferenzliste aus, die Ereignisbehandlung dazu erfolgt im PunkteReferenzController, welcher die Bezeichnung des Punktes ermittelt, und diese als Parameter der Methode punktLoeschen der Konstruktion übergibt. Da die Bezeichnung eindeutig ist, ist diese Methode imstande den entsprechenden Punkt aus der Liste aller geometrischen Objekte herauszusuchen, und ihn dann zu löschen. Dies geschieht folgendermaßen: Zuerst werden alle geometrischen Objekte gelöscht, die in der verbindungsListe referenziert sind. Das hat zur Folge das Mittelpunkte, Geraden und ähnliches mit nur einem Punkt nicht weiter existieren können. Zum Schluss wird der Punkt an sich gelöscht, in dem die Referenz auf diesen Punkt, in der Liste der geometrischen Objekte in der Konstruktion, einfach durch die Referenz auf den nachfolgenden Punkt der Liste ersetzt wird.

3.1.9 „Punkt verschieben“

Man muss im Programmmodus Punkt-Verschieben sein um dies durchführen zu können.

Der Benutzer wählt einen Punkt aus der Objektreferenzliste aus, die Ereignisbehandlung dazu erfolgt im PunkteReferenzController, welcher die Bezeichnung des Punktes ermittelt. Nun wählt der Benutzer im KonstruktionsView einen Ort, an den der Punkt verschoben werden soll. Die Koordinaten werden im KonstruktionsController ermittelt. Die Daten beider Punkte werden nun der Funktion punktVerschieben, der Klasse Konstruktion, übergeben. Als erster Parameter die Bezeichnung und als zweiter Parameter die Koordinaten der neu gewählten Position. In der Methode punktVerschieben der Konstruktion werden nun die Koordinaten des Punktes aus der Liste der geometrischen Objekte, auf den die Bezeichnung zutrifft, auf die Werte gesetzt, die der Methode als zweiter Parameter übergeben wurden.

3.1.10 Abstand zwischen zwei Punkten Berechnen

Man muss im Programmmodus Punkte-Abstand sein um dies durchführen zu können.

Der Benutzer wählt auf eine der beiden bekannten Arten ein oder zwei neue Punkte aus. Die entsprechende Auswertung der Ereignisse wird durch den PunkteReferenzController durchgeführt. Somit werden auf

GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



die eine oder andere Art zwei Punktkoordinaten ermittelt, die als Parameter der Methode `punkteAbstand` übergeben werden. Diese Methode berechnet dann den Abstand zwischen den Beiden Punkten, und gibt ihn an die Stelle, an der sie aufgerufen wurde, zurück. Also schlussendlich zurück an den `HauptController`, der den Abstand dann für den Benutzer ausgibt.

3.1.11 „Farbänderung“

Man muss im Programmmodus Punkte-Färben sein um dies durchführen zu können.

Es wird entweder zuerst der Farbwert gewählt, oder das geometrisches Objekt welches gefärbt werden soll. Alle dafür notwendigen Buttons werden beim Wechseln in den Farbmodus, durch den `HauptController` im Hauptfenster angelegt. Die Ereignisverarbeitung dieser Buttons erfolgt auch im Hauptfenster. Bei Änderung des Farbwertes wird ein kleines Fenster geöffnet, in welchem man die gewünschte Farbe auswählen kann. Wenn eine Farbe gewählt wird, so wird der Farbwert kurzzeitig gespeichert, und wenn dann auch das zu ändernde geometrische Objekt in der Objektreferenzliste ausgewählt wurde, kann die Farbänderung erfolgen. Dies erfolgt über die Methode `objektFarbeAendern` der Konstruktion, sowohl der neue Farbwert, als auch eine Referenz auf das zu ändernde Objekt übergeben werden.

Bemerkung: Bei der Instanzierung eines neuen geometrischen Objektes ist der Farbwert jedes geometrische Objektes auf `BLACK` gesetzt.

3.1.12 Programmhilfe

Man muss im Programmmodus Hilfe sein um dies durchführen zu können. Es wird ein Dialogfenster angezeigt, in dem sich der Benutzer die Funktionalität des Programms erklären lassen kann. Dazu wird eine Instanz der Klasse `Hilfe` erzeugt, welche im `HauptController` angelegt wird, wenn die Hilfefunktion das erste Mal aufgerufen wird. Der Benutzer kann zwischen verschiedenen Hilfetemen hin und herschalten. Die Ereignisverarbeitung wird vom Hilfefenster selbst verwaltet, da es auf keine anderen Teile des Programms Einfluss hat und nur sich selbst verwalten muss. Wenn einer der Buttons für ein anderes Hilfetema gewählt wird, dann wird ein anderer Hilfetext in die das Attribut `Hilfetext` geladen und angezeigt.

3.1.11 Koordinatenachsen/Gitternetz/Punktnamen ein- und ausblenden

Weiterhin ist es dem Nutzer möglich Koordinatenachsen, Gitternetz und/oder Punktnamen ein- und auszublenen. Dies Erfolgt über Buttons im Hauptfenster, und die Ereignisbehandlung dazu erfolgt im `HauptController`. Wenn eines der genannten Features aktiviert oder deaktiviert wird, dann wird eine entsprechende Variable im mit dem `HauptController` verbundenen `KonstruktionsView` auf wahr bzw. falsch gesetzt, und der `KonstruktionsView` angewiesen die Konstruktionsdarstellung zu aktualisieren, dies erfolgt auch über die Methode `konstruktionsAenderung` der Konstruktion, welche diesmal aber direkt vom `Controller` aufgerufen wird.

3.1.12 Änderung des Darstellungsbereiches des KonstruktionsView

Der Benutzer kann jederzeit durch drücken der Richtungstasten, den im `KonstruktionsView` sichtbaren Bereich der Konstruktion verschieben, oder mit der `+` oder `-` Taste in der Zeichenoberfläche zoomen. Die Ereignisverarbeitung der Tastatureingaben erfolgt im `HauptController`. Bei Drücken der Richtungstasten wird der Bereich, den die Zeichenoberfläche darstellt, verschoben, dazu benötigt man den Zeichenoberflächenmittelpunkt als Information an welcher Stelle sich dieser über der Konstruktion befindet. Beim Drücken der `+` bzw. `-` Taste wird das Attribut `Zoomfaktor`, des mit dem `HauptController` verbundenen `KonstruktionsView`, vergrößert oder verkleinert. Bei Neudarstellung der Views werden die veränderten Eigenschaften der verbundenen Views dann mit berücksichtigt und die Konstruktion dementsprechend in der Zeichenoberfläche dargestellt.

3.2.

Das Programm kann beendet werden, indem die erzeugte Instanz der Klasse `HauptController` gelöscht wird. Dies geschieht durch Anwenden der Methode `dispose`.

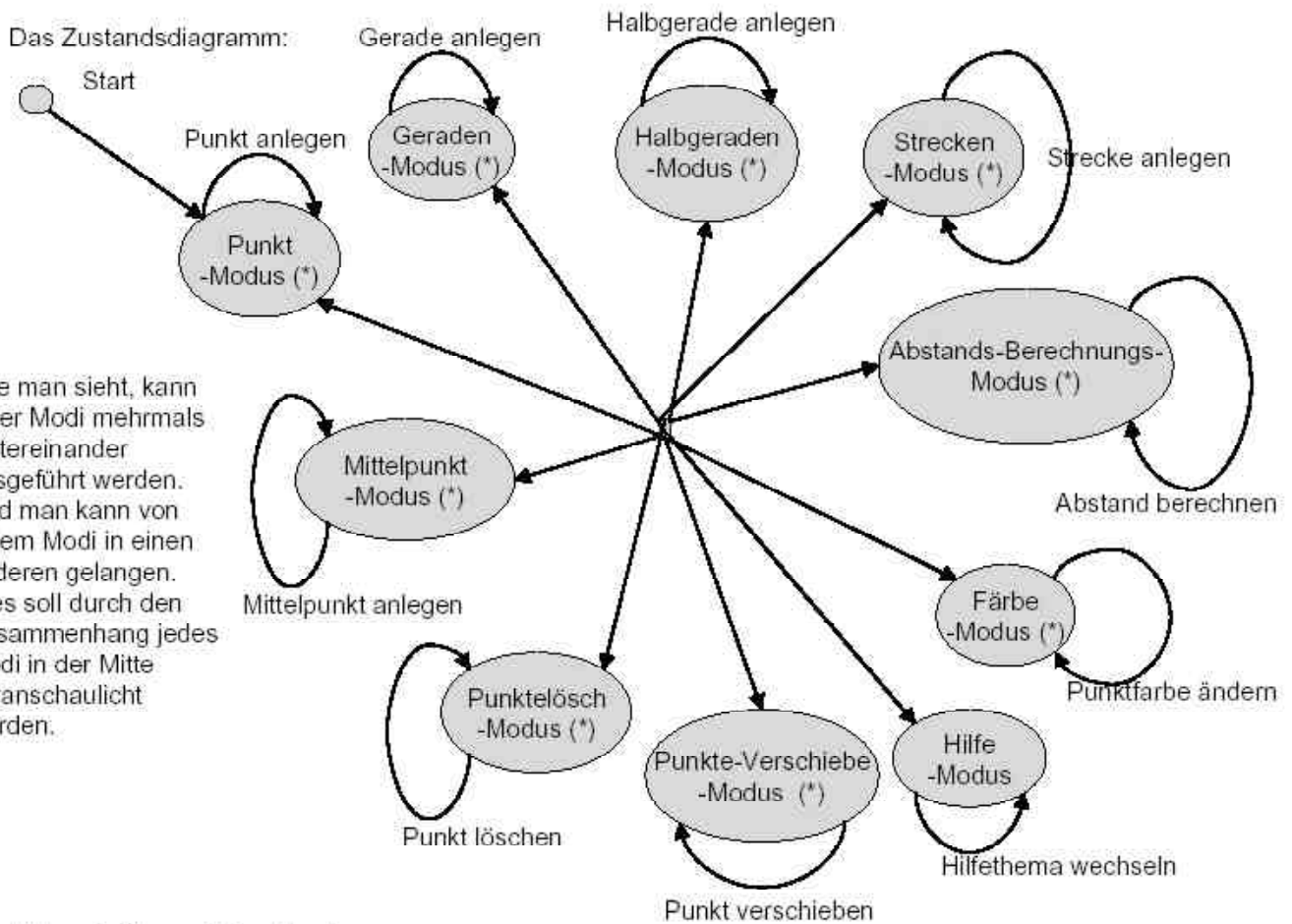
GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,

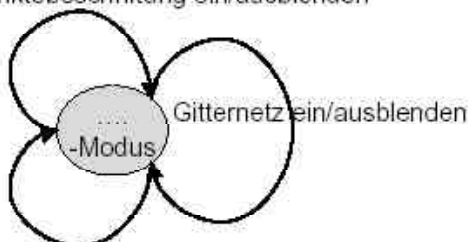


3.3.



Wie man sieht, kann jeder Modi mehrmals hintereinander ausgeführt werden. Und man kann von jedem Modi in einen anderen gelangen. Dies soll durch den Zusammenhang jedes Modi in der Mitte veranschaulicht werden.

Punktebeschriftung ein/ausblenden



Koordinatenachsen ein/ausblenden

Für alle Modi die mit einem (*) markiert sind gelten zudem diese Zustands Änderung. Diese wurde zur Erhaltung der Übersichtlichkeit nicht direkt eingezeichnet.

GEO01 - SOFTWARE SOLUTIONS

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

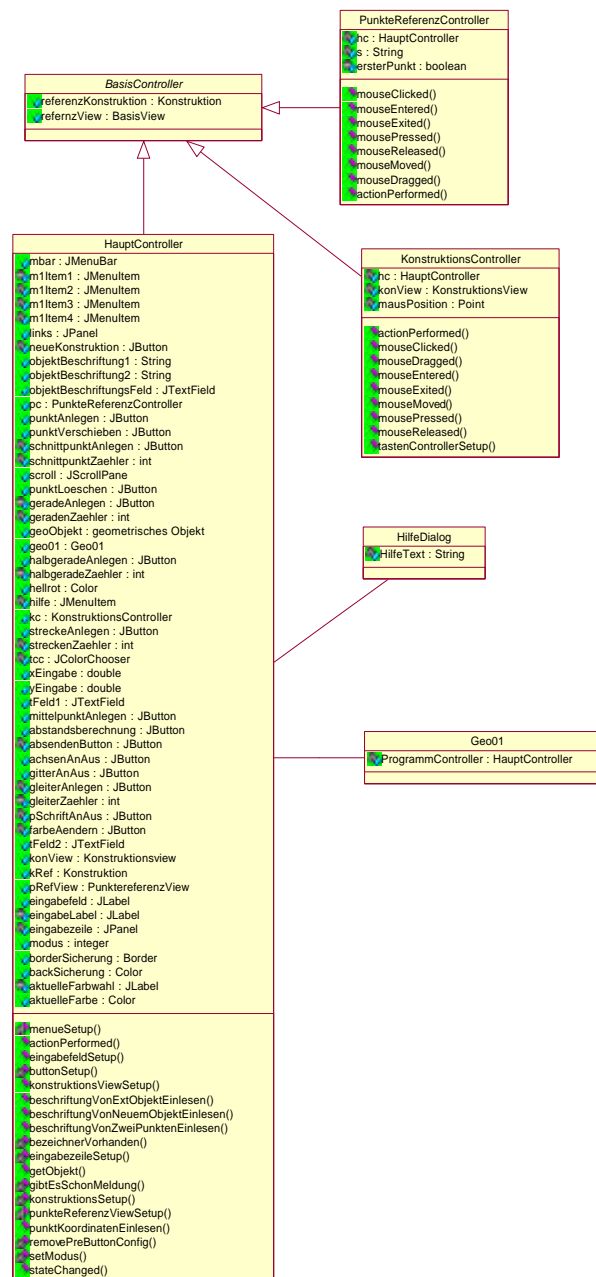
Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



4. Paket- und Klassenstrukturen

Darstellung als UML Diagramm erstellt mit Rational Rose

Package: Controller



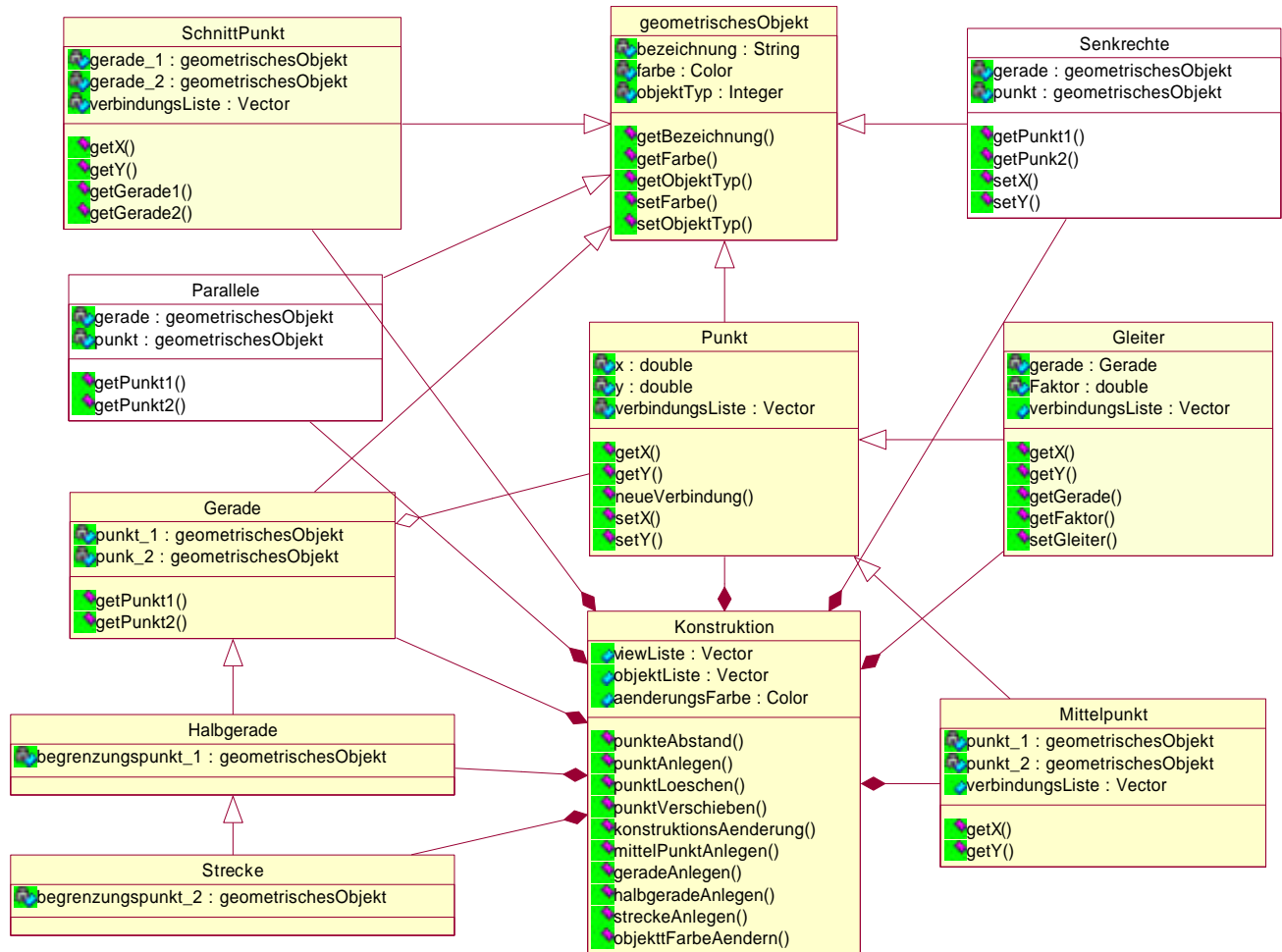
GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



Package: Modell



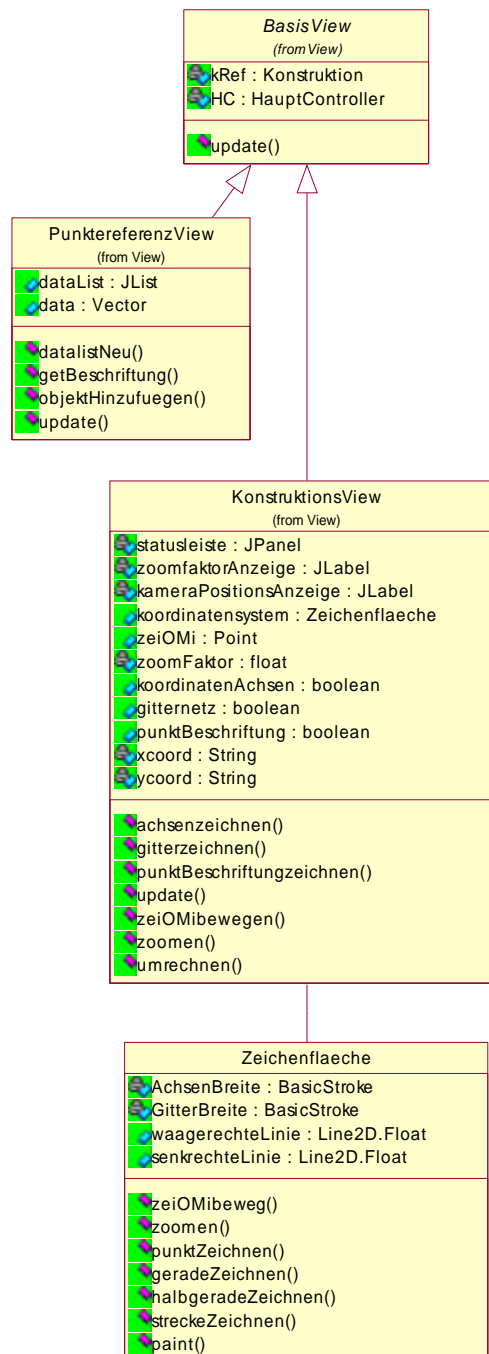
GEO01 - SOFTWARE SOLUTIONS

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

Erstellt von: Anne Güpner, Marcel Kretschmann, Matthias Reusche,



Package: View



Bemerkung: In der angehängten Geo01.mdl Datei sind die einzelnen Klassen noch einmal durch Kommentare erklärt. Ebenso ist dort die gesamte Klassenstruktur einzusehen.

Projektleiter: Dominic Rose

Mitarbeiter: Anne Güpner, Madlen Hartmann, Marcel Kretschmann, Matthias Reusche, Uta Schulze, Daniel Seifarth