Designentwurf der Applikation „DGS“

Inhaltsverzeichnis

1. Allgemeines

1.1. Kurzcharakterisierung

Das Programm **DGS** ist eine dynamische, menügesteuerte, graphische Java-Applikation, mit der sich geometrische Objekte erzeugen und darstellen lassen. Es ist auch eine Appletversion verfügbar.

1.2. Systemvoraussetzungen

DGS ist als eigenständige Einzelplatzapplikation konzipiert, die über eine grafische Nutzerschnittstelle mit gängigen Fenster-Techniken mit Hilfe der Maus und Tastatur bedient wird.
DGS benötigt als Voraussetzungen zum Starten die Java JRE 1.4 sowie für die Applet-Version die VM und einen Webbrowser mit Javaunterstützung bzw. einen Applet-Viewer.

1.3. Beschreibung der Produktumgebung

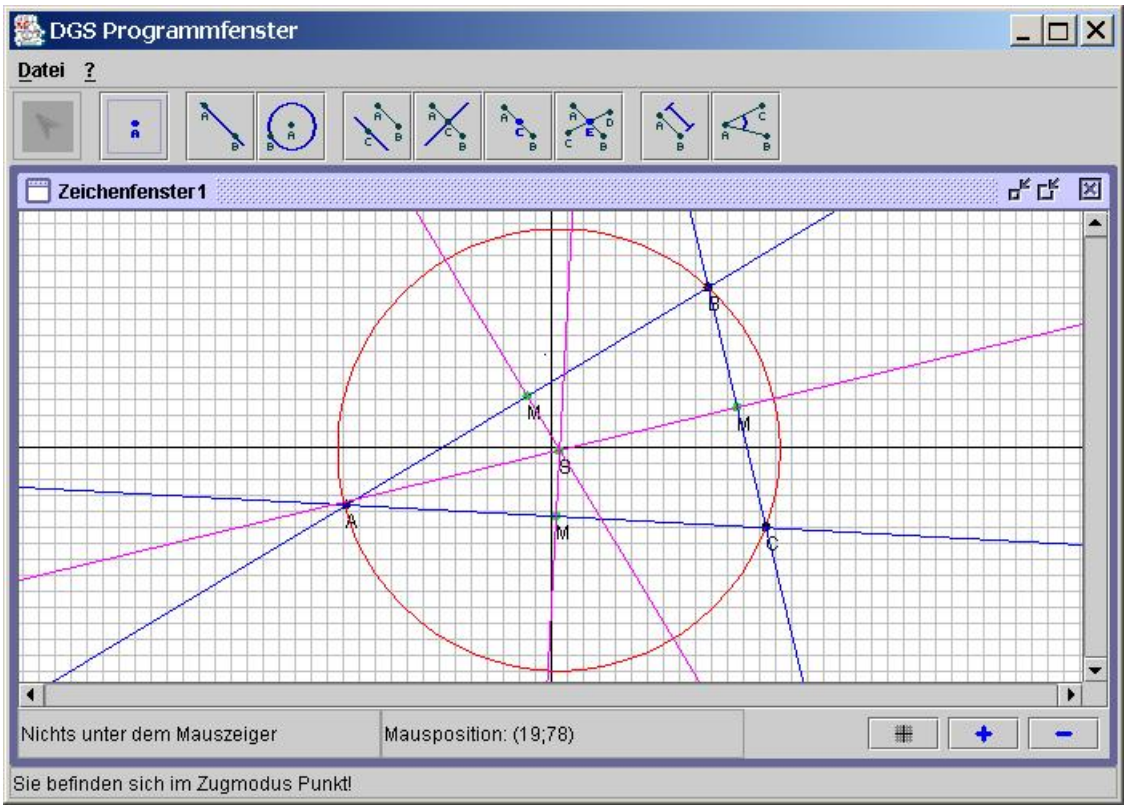
Die Datei "Geohelp.txt" enthält die Beschreibung der Hilfethemen und ist im ASCII-Format abgespeichert. Die Datei ist in verschiedene Absätze gegliedert, die jeweils ein spezielles Hilfethema enthalten. Dem Programm sind die Zeilennummern der Absätze bekannt, wodurch direkt zum gewünschten Thema gesprungen werden kann.
Die Datei "Geohelp.txt" sollte nicht verändert oder gelöscht werden, da sonst keine Hilfestellungen zum Programm mehr ausgegeben werden können.

1.4. Abgrenzung

- Das Programm ist nicht netzwerkfähig
- Kommt ohne Zugriff auf externe Programme aus
- Zur Verwendung der Hilfe muss die Datei Geohelp.txt im Paket enthalten sein.
- Keine 3D-Grafik

2. Produktübersicht

Die Applikation wird durch den Kommandozeilen-Aufruf
java DGS
im Wurzelverzeichnis der Pakethierarchie gestartet. Im Erfolgsfall wird das Hauptfenster aufgebaut.
Das ***Hauptfenster*** enthält am oberen Rand eine Menüleiste, eine Werkzeugleiste und in der Mitte eine Desktop-Fläche, in der im weiteren Verlauf die verschiedenen Zeichenflächen geöffnet werden können.
Über den Menüeintrag "*Datei*" und den Menüpunkt "*Neues Zeichenfenster*" wird ein neues Zeichenfenster erstellt.
Über den Menüeintrag "*Datei*" und den Menüpunkt "*Beenden*" werden alle Zeichenfenster geschlossen und das Programm beendet.
Über den Menüeintrag "*Datei*" und den Menüpunkt "*Schließen*" wird das aktuelle Zeichenfenster geschlossen.
Über den Menüeintrag „*?*“ und den Menüpunkt "*Hilfe*" wird die Hilfe aufgerufen – diese gibt das Hilfethema zur aktuell verwendeten Funktion aus.
Über den Menüeintrag "?" und den Menüpunkt "*Info*" kann eine Information über das Programm aufgerufen werden.
Unter der Menüleiste befindet sich die Toolbar, die die nachfolgenden Funktionen enthält:
"*Zugmodus Punkt*“, "*Zeichenmodus Punkt*“, "*Zeichenmodus Gerade*" , "*Zeichenmodus Kreis*“, *"Konstruktion Parallele"* , *"Konstruktion Senkrechte", "Konstruktion Mittelpunkt", "Konstruktion Schnittpunkt",* "*Abstand berechnen"* und *"Winkel berechnen"* . Die Funktionen in der Toolbar werden mit Hilfe anschaulicher Buttons in der Toolbar dargestellt.
In der Statusleiste des Hauptfensters wird angegeben, in welchem Modus man sich befindet(z.B. Zeichenmodus Punkt, Zugmodus Punkt etc.).



Wird ein Zeichenfenster ausgewählt, so wird für dieses die Toolbar aktiviert.

Das ***Zeichenfenster*** enthält am unteren Rand eine Statusleiste und in der Mitte die Zeichenfläche, in der im weiteren Verlauf die verschiedenen geometrischen Objekte konstruiert und gezeichnet werden können.
Es können mehre Zeichenfenster gleichzeitig angelegt werden, zwischen denen mit der Maus hin- und hergeschaltet werden kann. Die Zeichenfenster erhalten in der Reihenfolge wie sie geöffnet werden eine fortlaufende Nummerierung.
Über die Schaltflächen am Fensterrand kann die aktive Zeichenfläche minimiert, maximiert und geschlossen werden.
Die Statusleiste besteht aus drei Teilen. Im linken Teil werden aktuelle Meldungen zum Zeichenfenster ausgegeben (z.B. Weltkoordinaten eines Punktes). In der Mitte wird die aktuelle Mausposition angezeigt.
Der rechte Teil der Statusleiste enthält 3 Buttons:

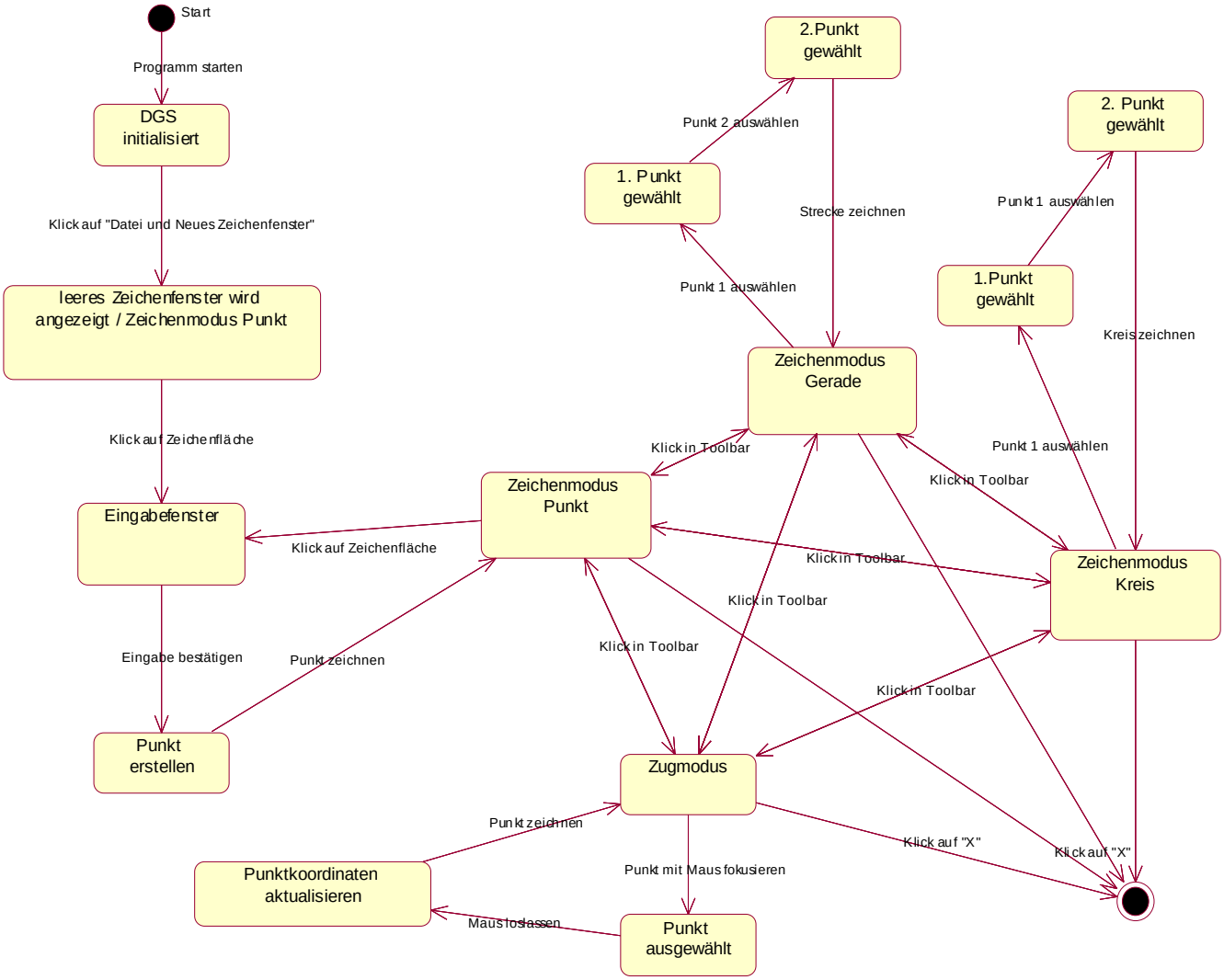
1. Button, um Koordinatensystem an- bzw. auszuschalten
2. Button, um die Skalierung im Zeichenfenster zu erhöhen
3. Button, um die Skalierung im Zeichenfenster zu verkleinern.

3. Grundsätzliche Design-Entscheidungen

3.1. Allgemeines

Dem Paket- und Klassen-Design liegen das Ereigniskonzept sowie der MVC-Ansatz zugrunde.
Das **MVC**-Modell überträgt die Trennung von Eingabe, Verarbeitung und Ausgabe auf die GUI-basierte Applikation. Dabei entspricht das **M**odell der Verarbeitung, der **V**iew der Ausgabe und der **C**ontroller der Eingabe.
Die **graphische Oberfläche** des **DGS** wurde mit Swing-Klassen, gängigen Fenstertechniken und dem Ereignis-Aktions-Konzept realisiert. Dabei wurde für die Grafikelemente auf Swing-Standardkomponenten zurückgegriffen. Die Steuerung des Programms erfolgt standardmäßig über die Maus, es werden aber auch Benutzereingaben über die Tastatur benötigt.

3.2. Zustandsdiagramm zum Programmablauf für die Zeichen- und Zugmodi



3.3. Darstellung geometrischer Objekte – die Modellsicht (Model)

Das Model enthält die mathematische Beschreibung aller beteiligten geometrischen Objekte. Diese sind Punkt, Gerade und Kreis. Das Model liefert den Zustand der Daten und reagiert auf Befehle, diesen Zustand zu ändern.

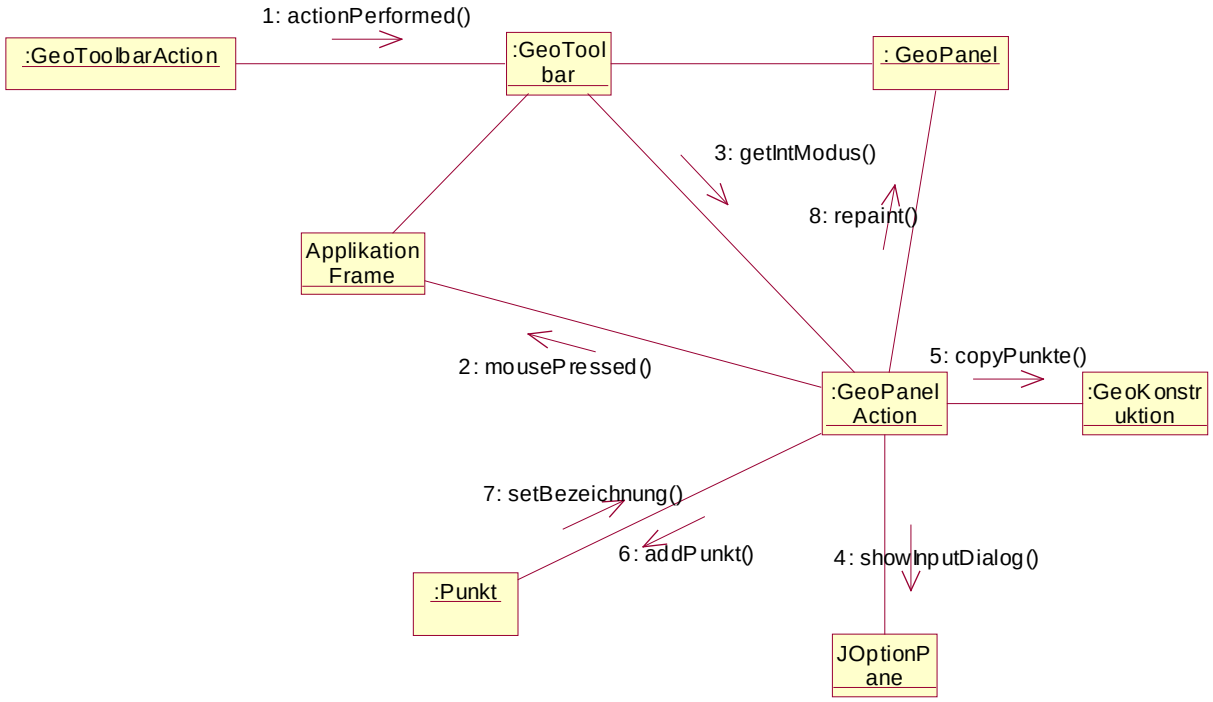
3.4. Darstellung geometrischer Objekte – die Visualisierung (View)

Zur Visualisierung muss einerseits zwischen den „Weltkoordinaten“ des Modells und den "Pixelkoordinaten" der Zeichenfläche umgerechnet werden. Andererseits müssen aus den Modellkoordinaten der geometrischen Objekte die Aufrufparameter der Grafikprimitive bestimmt werden, mit denen die zugehörige Visualisierung erzeugt werden soll.
Die Visualisierung wird hierbei von der Klasse **GeoPanel** übernommen.

3.5. Darstellung geometrischer Objekte - der Kontrollfluss (Controller)

Der Controller implementiert das Benutzerinterface und stellt somit die Funktionalität der Menüs, der Toolbar und der Dialoge zur Verfügung. Die möglichen Eingaben des Anwenders werden von dem Controller auf die verschiedenen Methoden des Models und des Views abgebildet. Der Controller stellt z.B. die Dialoge zum Erstellen geometrischer Objekte bereit (Punkt zeichnen, Gerade zeichnen etc.).

3.6. Kollaborationsdiagramm : Zeichnen eines Punktes



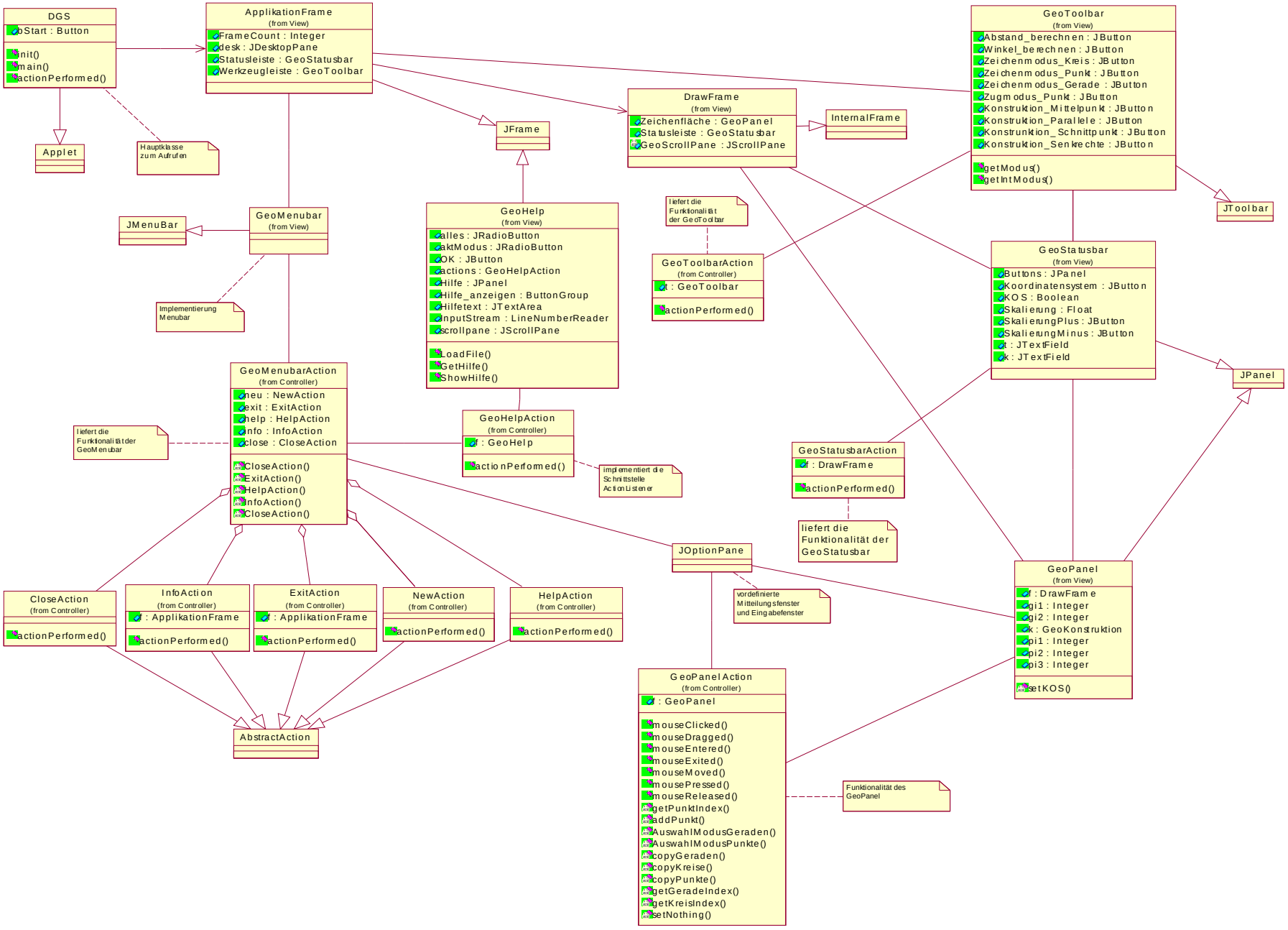
Nach einer festgelegten Aktion in der **GeoToolbar** wird mit der Methode *actionPerformed()* der **Zeichenmodus_Punkt** = true gesetzt. Wird nun im **GeoPanel** eine Aktion (Mausklick) ausgeführt, so wird die Methode *mousePressed()* ausgeführt. Diese holt sich den aktuellen Modus aus der **GeoToolbar**. Liefert die Methode *getIntModus()* 1 (für Zeichenmodus_Punkt) zurück, wird ein **JOptionPane** mit der Methode *showInputDialog()* ausgeführt. Im nächsten Schritt wird die Methode *copyPunkte()* ausgeführt. Diese kopiert das Punktfeld aus **GeoKonstruktion** in ein um 1 vergrößertes Feld. Danach wird ein neuer Punkt mit den Mauskoordinaten mit Hilfe der Funktion *addPunkt()* initialisiert.Von diesem neuen Punkt wird die Methode *setBezeichnung()* aufgerufen. Der nun erhaltene neue Punkt wird in dem Feld in **GeoKonstruktion** hinzugefügt, dessen Methode *repaint()* im Anschluss den neu erhaltenen Punkt in **GeoPanel** visualisiert.

4. Paket- und Klassenstruktur

Die Applikation ist in die Hauptklasse **DGS** und die 3 Pakete **Model**, **View** und **Controller** aufgeteilt.

4.1. Überblick der Klassenstruktur

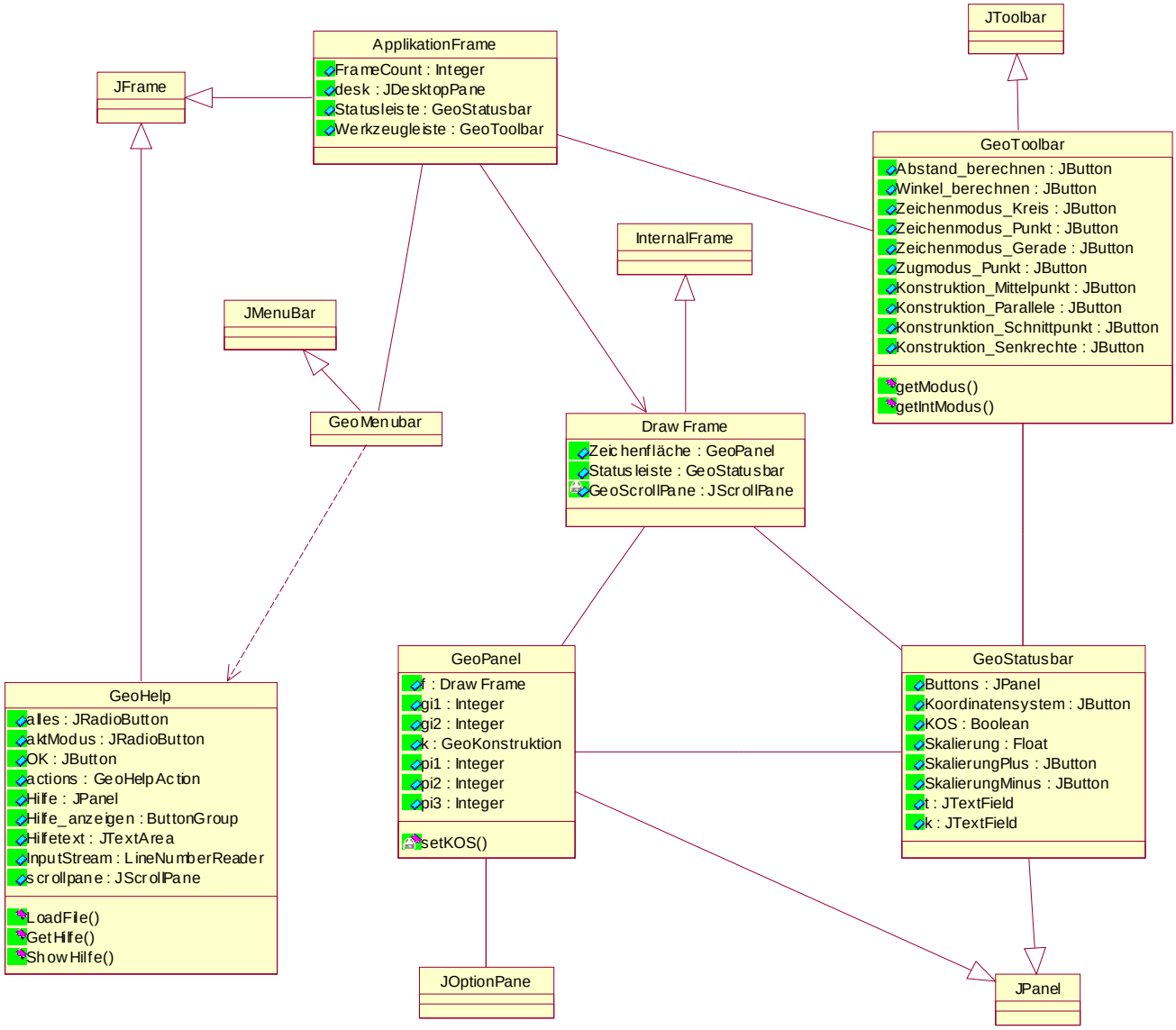
à auf der nächsten Seite


DGS
boStart : Button
init()
main()
actionPerformed()
Applet
Hauptklasse zum Aufrufen
ApplikationFrame
(from View)
FrameCount : Integer
desk : JDesktopPane
Statusleiste : GeoStatusbar
Werkzeugleiste : GeoToolbar
JFrame
DrawFrame
(from View)
Zeichenfläche : GeoPanel
Statusleiste : GeoStatusbar
GeoScrollPane : JScrollPane
InternalFrame
GeoToolbar
(from View)
Abstand_berechnen : JButton
Winkel_berechnen : JButton
Zeichenmodus_Kreis : JButton
Zeichenmodus_Punkt : JButton
Zeichenmodus_Gerade : JButton
Zugmodus_Punkt : JButton
Konstruktion_Mittelpunkt : JButton
Konstruktion_Parallele : JButton
Konstrunktion_Schnittpunkt : JButton
Konstruktion_Senkrechte : JButton
getModus()
getIntModus()
JToolbar
JMenuBar
GeoMenubar
(from View)
Implementierung Menubar
GeoHelp
(from View)
alles : JRadioButton
aktModus : JRadioButton
OK : JButton
actions : GeoHelpAction
Hilfe : JPanel
Hilfe_anzeigen : ButtonGroup
Hilfetext : JTextArea
inputStream : LineNumberReader
scrollpane : JScrollPane
LoadFile()
GetHilfe()
ShowHilfe()
liefert die Funktionalität der GeoToolbar
GeoToolbarAction
(from Controller)
t : GeoToolbar
actionPerformed()
GeoStatusbar
(from View)
Buttons : JPanel
Koordinatensystem : JButton
KOS : Boolean
Skalierung : Float
SkalierungPlus : JButton
SkalierungMinus : JButton
t : JTextField
k : JTextField
JPanel
GeoMenubarAction
(from Controller)
neu : NewAction
exit : ExitAction
help : HelpAction
info : InfoAction
close : CloseAction
CloseAction()
ExitAction()
HelpAction()
InfoAction()
CloseAction()
liefert die Funktionalität der GeoMenubar
GeoHelpAction
(from Controller)
f : GeoHelp
actionPerformed()
implementiert die Schnittstelle ActionListener
GeoStatusbarAction
f : DrawFrame
actionPerformed()
liefert die Funktionalität der GeoStatusbar
JOptionPane
vordefinierte Mitteilungsfenster und Eingabefenster
GeoPanel
(from View)
f : DrawFrame
gi1 : Integer
gi2 : Integer
k : GeoKonstruktion
pi1 : Integer
pi2 : Integer
pi3 : Integer
setKOS()
CloseAction
(from Controller)
actionPerformed()
InfoAction
(from Controller)
f : ApplikationFrame
actionPerformed()
ExitAction
(from Controller)
f : ApplikationFrame
actionPerformed()
NewAction
(from Controller)
actionPerformed()
HelpAction
(from Controller)
actionPerformed()
AbstractAction
GeoPanelAction
(from Controller)
f : GeoPanel
mouseClicked()
mouseDragged()
mouseEntered()
mouseExited()
mouseMoved()
mousePressed()
mouseReleased()
getPunktIndex()
addPunkt()
AuswahlModusGeraden()
AuswahlModusPunkte()
copyGeraden()
copyKreise()
copyPunkte()
getGeradeIndex()
getKreisIndex()
setNothing()
Funktionalität des GeoPanel

4.2. Die Hauptklasse DGS

Die Hauptklasse **DGS** enthält die main-Methode und ist für die Initialisierung der Applikation verantwortlich.

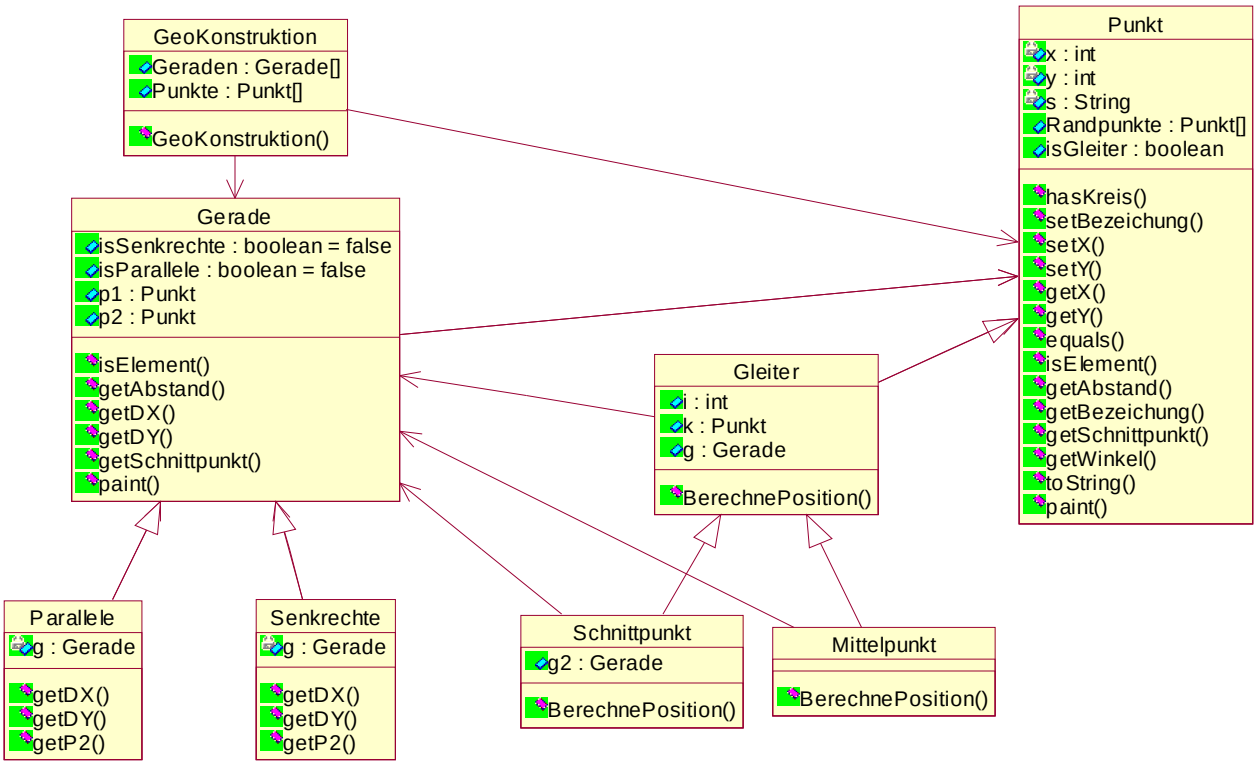
4.3. Das Paket View



Dieses Paket enthält alle GUI-Elemente (z.B. Fensterklassen, Toolbar, etc.) des Programms.
Die zentrale Klasse ist **ApplikationFrame**, welche die Swing-Klasse ***JFrame*** erweitert und von der im Laufe des Programms eine Instanz, das Startfenster der Applikation, angelegt wird. Dessen Attribut **FrameCount** enthält die Anzahl der momentan instanziierten Zeichenfenster.
Die Menüleiste **GeoMenubar** wird durch eine abgeleitete Klasse von ***JMenuBar*** und mit Hilfe der Klassen ***JMenu*** und ***AbstractAction*** angelegt und danach vom Konstruktor des **ApplikationFrame** initialisiert. Über diese Menüpunkte können weitere Fenster, wie z.B. Zeichenfenster, Hilfefenster und ein Infofenster angezeigt werden, wobei das Infofenster direkt mit der Methode *showMessageDialog()* der Klasse ***JOptionPane*** erstellt wird. Das Hilfefenster, welches durch die Klasse **GeoHelp** erzeugt wird, besteht lediglich aus einem ***JButton***, ***JRadioButton*** und einem ***JTextField***, in welches der Hilfetext aus der ASCII-Hilfedatei "Geohelp.txt" in Abhängigkeit des Zustandes der **GeoToolbar** angezeigt wird.

In der Klasse **GeoToolbar** werden Schaltflächen mit aussagefähigen Bildern angezeigt, mit deren Hilfe der Benutzer den aktuellen Modus auswählen kann.
Die Verwaltung der einzelnen Zeichenflächen erfolgt direkt mit der Swing-Klasse ***JDesktopPane*** für den Hintergrund desjenigen Bereichs im Hauptbildschirm, in dem die Zeichenflächen dargestellt werden.
Das Zeichenfenster selbst, welches in der Klasse **DrawFrame** realisiert wird, ist eine abgeleitete Klasse von ***JInternalFrame*** und enthält die 2 Panels: **GeoPanel** und **GeoStatusbar**.
Die Klasse **GeoPanel** wird für die Darstellung der in den Feldern **Geraden** und **Punkte**, der sich in der Klasse **GeoKonstruktion** befindlichen geometrischen Objekte, verwendet. Die Ausgabe wird durch die Methode *paint(Graphics g)* aus der Oberklasse ***JPanel*** realisiert. Um jedes Objekt einzeln zeichnen zu können, müssen die **Punkte** vom Weltkoordinatensystem in das benötigte Gerätekoordinatensystem in Abhängigkeit der **Skalierung** in der **GeoStatusbar** transformiert.
Die Klasse **GeoStatusbar** dient für das Anzeigen von Status-Mitteilungstexten im **ApplikationFrame** und für die Interaktion mit dem Benutzer beim Modellieren von geometrischen Objekten im **DrawFrame.** Im DrawFrame enthält die **GeoStatusbar** außerdem Schaltflächen, die die Skalierung der Zeichenfläche verändern und das Koordinatensystem ein bzw. ausblenden und ein weiteres **JTextfield** zur Anzeige der aktuellen Mauskoordinaten.

4.4. Das Paket Model



Das Model jeder Konstruktion ist in einer Instanz der Klasse **GeoKonstruktion** im **DrawFrame** gespeichert. Diese Klasse enthält ein Feld für Punkte und eins für Geraden.
Die Klasse **Punkt** dient der mathematischen Beschreibung von Punkten und enthält die Koordinaten, eine Bezeichnung sowie ein Feld für die Randpunkte der Kreise um diesen Punkt.
In der Klasse gibt es Methoden zum:

- Setzen und Auslesen der gespeicherten Koordinaten und der Bezeichnung des Punktes
- Vergleichen
- Berechnen von Abständen und Winkeln zwischen Punkten
- Berechnen des Schnittpunktes zwischen einem Kreis mit der Strecke zwischen dem Mittelpunkt des Kreises und einem weiteren Punkt (*getSchnittpunkt()*)
- Überprüfung, ob ein Punkt auf dem Rand eines Kreises liegt (*isElement()*)

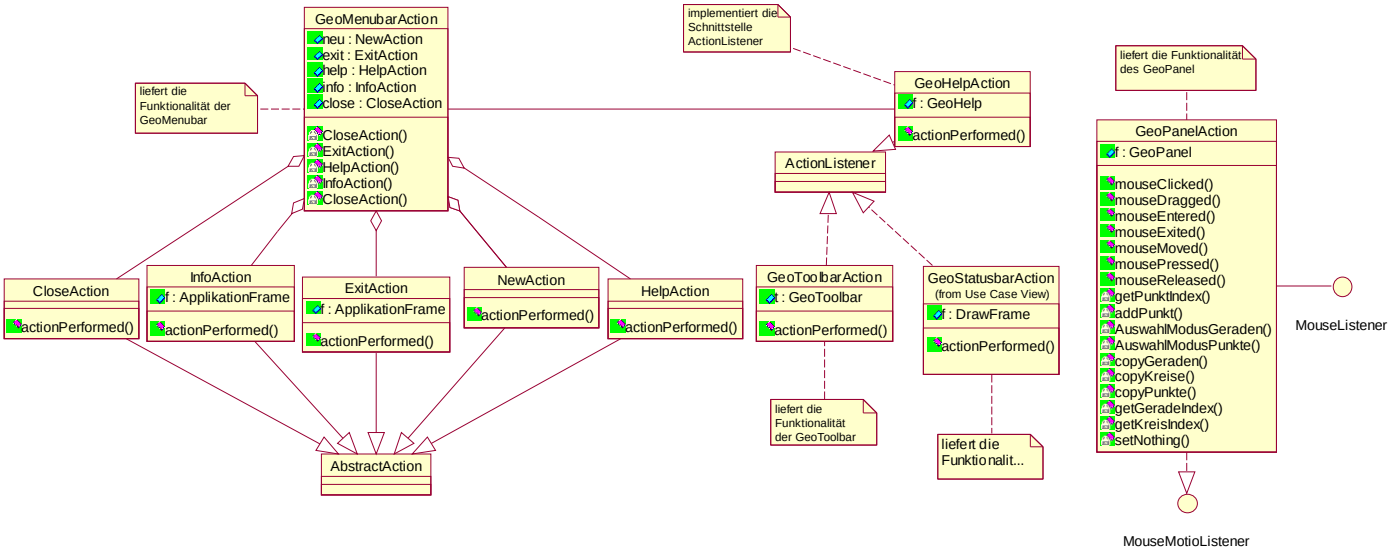
Die Klasse **Gleiter** ist von **Punkt** abgeleitet. Neben den Methoden und Attributen der Oberklasse werden noch Referenzen auf den Kreis bzw. die Gerade und die Methode *BerechnePosition()* hinzugefügt. Mit dieser Methode wird die Position des Gleiters auf dem Kreis/Gerade neu berechnet.

Die Unterklassen von Gleiter sind Mittelpunkt und Schnittpunkt. Bei der Klasse Schnittpunkt wird lediglich die Referenz auf eine zweite Gerade hinzugefügt. Die Methode *BerechnePosition()* aus **Gleiter** wird geeignet überschrieben.

Für das Modellieren einer **Gerade** werden 2 Punkte aus dem Feld **Punkte**, der Klasse **GeoKonstruktion**, benötigt, so dass die Gerade eindeutig durch 2 Punkte bestimmt ist. Zusätzlich werden Funktionen zum Berechnen von Schnittpunkten und Abständen sowie zur Bestimmung des Richtungsvektors zur Verfügung gestellt.

Die Unterklassen von Gerade sind Senkrechte und Parallele. Die Methode *getP2()* liefert einen virtuellen Punkt zum Zeichnen der Senkrechten/Parallelen.

4.5. Das Paket Controller



Das Paket enthält die zentralen Klassen **GeoMenubarAction**, **GeoToolbarAction** und **GeoPanelAction**.
Die Klasse **GeoMenubarAction** stellt die nachfolgenden Funktionalitäten des Hauptfenstermenüs zur Verfügung:

- die Methode *Neu()* , die ein neues Zeichenfenster erstellt
- die Methode *Close()* , die das ausgewählte Zeichenfenster schließt
- die Methode *Exit()* , die das Programm beendet
- die Methode *Help()* , die das Hilfefenster aufruft
- die Methode *Info()*, die eine kurze Information über das Programm ausgibt

Die Klasse **GeoToolbarAction** steuert die Funktionalität der **GeoToolbar**, indem der entsprechende Button für den aktuellen Modus deaktiviert wird und geeignete Texte in der **GeoStatusbar** angezeigt werden. Die Methode *getIntModus()/getModus()* gibt je nach gewählten Modus einen Integerwert/String zurück.
Die Klasse **GeoPanelAction** steuert die Ereignisse in der Zeichenfläche.

Weiterhin ist die Methode *addPunkt()* implementiert, die nach einer Mausaktion auf die Zeichenfläche zum Feld **Punkte** in **GeoKonstruktion** einen Punkt/Gleiter hinzufügt.
Die Methoden *getPunktIndex(), getGeradenIndex() und getKreisIndex()* identifizieren durch das zurückgegebene Mausereignis auf der Zeichenfläche das geometrische Objekt in den beiden Feldern in **GeoKonstruktion**, welches sich unter dem Mauszeiger befindet.
Die Methoden *copyPunkte(), copyGeraden()* und *copyKreise()* kopieren die vorhandenen jeweiligen geometrischen Objekte in ein um 1 vergrößertes Feld, in dessen letzten uninitialisierten Platz danach ein neues geometrisches Objekt eingefügt werden kann.
Mit den beiden Methoden *AuswahlModusPunkte()* und *AuswahlModusGeraden()* wird die Auswahl von mehreren geometrischen Objekten in der Zeichenfläche ermöglicht.

Die Klasse *GeoHelpAction* implementiert die Funktionen für die Buttons im Hilfefenster. Klickt der Benutzer auf den Radiobutton *aktModus* wird die Hilfe für den in der GeoToolbar gewählten Modus durch die Methode *getHilfe()* aus **GeoHelp** ausgewählt und durch *ShowHilfe()* angezeigt. Beim Klicken auf den Radiobutton „*alles*“ wird durch die Methode *ShowHilfe()* aus **GeoHelp** die gesamte Hilfedatei angezeigt. Ein Klick auf OK schließt das Hilfefenster.

Die Klasse **GeoStatusbarAction** implementiert die Funktionen der Buttons der GeoStatusbar.
Möchte der Benutzer die Skalierung ändern, so klickt er auf den Plusbutton zum Erhöhen bzw. auf den Minusbutton zum Erniedrigen, wobei durch dieses Ereignis die Methode *setSkalierung()* aufgerufen wird, welche die **Skalierung** entsprechend anpasst. Möchte der Benutzer das Koordinatensystem aus bzw. einblenden, klickt er auf den entsprechenden Button in der Statusleiste. Dazu wird eine Flag gesetzt und die Methode *setKOS()* aufgerufen.