Klassen- und Methodenübersicht

Im Folgenden wird eine Übersicht über die grundlegenden Klassen und Methoden der GLOOP Bibliothek (Version 3.7) gegeben. Die folgende Auflistung ist nicht vollständig und umfasst lediglich die wichtigsten Klassen und Methoden. Alternativ steht eine *Komplettübersicht* der GLOOP Bibliothek zur Verfügung.

1 Unterklassen von GLObjekt

1.1 Klasse GLKegel (Oberklasse GLObjekt)

Konstr. **`GLKegel(double pX, double pY, double pZ, double pRadius, double pHoehe)`**
Erstellt einen Kegel mit der Höhe `pHoehe` und dem Grundflächenradius `pRadius`.

1.2 Klasse GLKegelstumpf (Oberklasse GLObjekt)

Konstr. **`GLKegelstumpf(double pX, double pY, double pZ, double pRadius1, double pRadius2, double pHoehe)`**
Erstellt einen Kegelstumpf mit der Höhe `pHoehe` und den Radien `pRadius1` und `pRadius2`.

1.3 Klasse GLKugel (Oberklasse GLObjekt)

Konstr. **`GLKugel(double pX, double pY, double pZ, double pRadius)`**
Erstellt an der Stelle `(pX,pY,pZ)` eine Kugel mit dem Radius `pRadius`.

1.4 Klasse GLLicht (Oberklasse GLObjekt)

Konstr. **`GLLicht()`**
Erstellt eine weiße Lichtquelle an der Stelle `(-10000,10000,10000)`.

1.5 Klasse GLQuader (Oberklasse GLObjekt)

Konstr. **`GLQuader(double pX, double pY, double pZ, double pLX, double pLY, double pLZ)`**
Erstellt einen Quader mit den Abmessungen `pLX`, `pLY`, `pLZ` bzgl. der drei Raumdimensionen.

1.6 Klasse GLTafel (Oberklasse GLObjekt)

Konstr. **`GLTafel(double pX, double pY, double pZ, double pLX, double pLY)`**
Erstellt eine zweidimensionale, beschriftbare Tafel in der Szene.

Anfrage **`String gibText()`**
Liefert den Schriftzug auf der Tafel.

Auftrag **`void setzeAutodrehung(boolean pD)`**
Schaltet die automatische Ausrichtung der Tafel zur Kamera ein bzw. aus.

Auftrag **`void setzeKamerafixierung(boolean pF)`**
Schaltet die Fixierung der Tafel im Kamerabild ein bzw. aus.

Auftrag **`void setzeText(String pText, double pGroesse)`**
Setzt einen Schriftzug in der angegebenen Größe auf die Tafel.

1.7 Klasse GLTorus (Oberklasse GLObjekt)

Konstr. **`GLTorus(double pX, double pY, double pZ, double pRadius, double pDicke)`**
Erstellt einen Torus mit dem Radius `pRadius` und der Dicke `pDicke`.

1.8 Klasse GLWuerfel (Oberklasse GLObjekt)

Konstr. **`GLWuerfel(double pX, double pY, double pZ, double pSeitenlaenge)`**
Erstellt einen Würfel mit der Seitenlänge `pSeitenlaenge`.

1.9 Klasse GLZylinder (Oberklasse GLObjekt)

Konstr. **`GLZylinder(double pX, double pY, double pZ, double pRadius, double pHoehe)`**
Erstellt einen Zylinder mit der Höhe `pHoehe` und mit dem Radius `pRadius`.

2 Methoden der Oberklasse GLObjekt

Auftrag **`void drehe(double pWX, double pWY, double pWZ)`**
Dreht das Objekt um durch den Mittelpunkt des Objektes gehende Parallelen der Koordinatenachsen.

Auftrag **`void drehe(double pWX, double pWY, double pWZ, double pX, double pY, double pZ)`**
Dreht das Objekt um durch den Punkt `(pX,pY,pZ)` gehende Parallelen der Koordinatenachsen.

Anfragen **`double gibX()`**
`double gibY()`
`double gibZ()`
Liefert die entsprechende Koordinate des Mittelpunktes des Objekts.

Auftrag **`void loesche()`**
Löscht das Objekt.

Auftrag **`void setzeDrehung(double pWX, double pWY, double pWZ)`**
Dreht das Objekt um durch den Mittelpunkt des Objektes gehende Parallelen der Koordinatenachsen, unabhängig von der vorangegangenen Ausrichtung des Objektes, auf die angegebenen Drehwinkel.

Auftrag **`void setzeFarbe(double pR, double pG, double pB)`**
Setzt die Farbe des Objektes. `pR` = Rotanteil, `pG` = Grünanteil, `pB` = Blauanteil.

Auftrag **`void setzePosition(double pX, double pY, double pZ)`**
Setzt die Position des Objekts auf die Position `(pX,pY,pZ)`.

Auftrag **`void setzeSkalierung(double pG)`**
Absolute Variante von `skaliere`.

Auftrag **`void setzeSkalierung(double pX, double pY, double pZ)`**
Absolute Variante von `skaliere`.

Auftrag **`void setzeTextur(GLTextur pTex)`**
Überzieht das Objekt mit der übergebenen Textur.

Auftrag **`void setzeTextur(String pDateiname)`**
Erstellt aus einer Datei ein Texturobjekt und überzieht das Objekt mit dieser Textur.

Auftrag **`void skaliere(double pG)`**
Verändert die Größe des Objektes um den Faktor `pG`.

Auftrag **`void skaliere(double pX, double pY, double pZ)`**
Verändert die Größe des Objektes in Richtung jeder Achse um einen separaten Wert.

Auftrag **`void verschiebe(double pX, double pY, double pZ)`**
Verschiebt das Objekt entlang der drei Koordinatenachsen.

3 Weitere Grafikklassen

3.1 Klasse GLBoden (Oberklasse Object)

Konstr. **`GLBoden(GLTextur pBoden)`**
`GLBoden(String pBoden)`
Erstellen eine endlose Ebene in der Szene, die mit der im Parameter übergebenen Textur bzw. Bilddatei gekachelt ist.

Auftrag **`void loesche()`**
Löscht das Objekt.

3.2 Klasse GLHimmel (Oberklasse Object)

Konstr. **`GLHimmel(GLTextur pHimmel)`**
`GLHimmel(String pHimmel)`
Erstellt eine Himmelssphäre, die auf der Innenseite die im Parameter übergebene Textur bzw. Bilddatei zeigt.

Auftrag **`void loesche()`**
Löscht das Objekt.

3.3 Klasse GLNebel (Oberklasse Object)

Konstr. **`GLNebel()`**
Erstellt ein Nebelobjekt, das die Szene mit gleichmäßigem Dunst ausfüllt.

Auftrag **`void loesche()`**
Entfernt den Nebel aus der Szene.

Auftrag **`void setzeFarbe(double pR, double pG, double pB)`**
Setzt die Farbe des Nebels.

Auftrag **`void setzeNebelbereich(double pAnfang, double pEnde)`**
Der Nebelbereich wird gesetzt.

4 Verschiedene Kameraklassen

4.1 Klasse GLKamera (Oberklasse Object)

Konstr. **`GLKamera()`**
Erstellt eine Kamera im Vollbildmodus.
`GLKamera(int pB, int pH)`
Erstellt eine Kamera, deren Fenster die Breite `pB` und die Höhe `pH` hat.

Anfragen **`double gibBlickpunktX()`**
`double gibBlickpunktY()`
`double gibBlickpunktZ()`
Gibt die entsprechende Komponente des Blickpunktes der Kamera zurück.

Anfragen **`double gibX()`**
`double gibY()`
`double gibZ()`
Gibt die entsprechende Koordinate der Position der Kamera zurück.

Auftrag **`void schwenkeHorizontal(double pWinkel)`**
Dreht die Kamera in der Art eines Horizontalschwenks (links / rechts) um den Winkel `pWinkel`.

Auftrag **`void schwenkeVertikal(double pWinkel)`**
Dreht die Kamera in der Art eines Vertikalschwenks (oben / unten) um den Winkel `pWinkel`.

Auftrag **`void setzeBlickpunkt(double pX, double pY, double pZ)`**
Setzt den Blickpunkt der Kamera auf den Punkt `(pX,pY,pZ)`.

Auftrag **`void setzePosition(double pX, double pY, double pZ)`**
Setzt die Position der Kamera auf den Punkt `(pX,pY,pZ)`.

Auftrag **`void verschiebe(double pX, double pY, double pZ)`**
Verschiebt die Kamera um den Wert `pX` auf der X-Achse, `pY` auf der Y-Achse und `pZ` auf der Z-Achse.

Auftrag **`void vor(double pWeite)`**
Lässt die Kamera in Richtung des Blickpunktes um `pWeite` vorfahren.

4.2 Klasse GLEntwicklerkamera (Oberklasse GLSchwenkkamera)

Konstr. **`GLEntwicklerkamera()`**
Erstellt eine Entwicklerkamera im Vollbildmodus.
`GLEntwicklerkamera(int pB, int pH)`
Erstellt eine Entwicklerkamera, deren Kamerafenster die Breite `pB` und die Höhe `pH` hat.

5 Eingabeklassen

5.1 Klasse GLTastatur (Oberklasse Object)

Konstr. **`GLTastatur()`**
Erstellt ein neues Tastaturobjekt.

Anfrage **`boolean alt()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean backspace()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean enter()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean esc()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean istGedrueckt()`**
Liefert `true`, wenn irgendeine Taste gedrückt ist.

Anfrage **`boolean istGedrueckt(char pT)`**

Liefert `true`, wenn die dem Zeichen `pT` entsprechende Taste gedrückt ist.

Anfrage **`boolean links()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean oben()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean rechts()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean shift()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean strg()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean tab()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

Anfrage **`boolean unten()`**
Liefert `true`, wenn die angefragte Taste gedrückt ist.

5.2 Klasse GLMaus (Oberklasse Object)

Konstr. **`GLMaus()`**
Erstellt ein neues Mausobjekt.

Anfrage **`boolean doppelklick()`**
Liefert `true`, wenn ein Doppelklick durchgeführt wurde.

Anfrage **`boolean gedruecktLinks()`**
Liefert `true`, wenn die linke Maustaste gerade gedrückt ist.

Anfrage **`boolean gedruecktRechts()`**
Liefert `true`, wenn die rechte Maustaste gerade gedrückt ist.

Anfrage **`int gibX()`**
`int gibY()`
Liefert die entsprechende Mauskoordinate im Kamerafenster.

Anfrage **`boolean linksklick()`**
Liefert `true`, wenn ein Linksklick durchgeführt wurde.

Anfrage **`boolean rechtsklick()`**
Liefert `true`, wenn ein Rechtsklick durchgeführt wurde.

6 Hilfsklassen

6.1 Klasse Sys (Oberklasse Object)

Auftrag **`static void beenden()`**
Beendet das aktuelle Programm.

Auftrag **`static void erstelleAusgabe(String pM)`**
Gibt den String `pM` auf einer am unteren Bildrand eingeblendeten Konsole aus.

Auftrag **`static void erstelleAusgabe(String pT, String pM)`**
Gibt den String `pM` auf einer am unteren Bildrand eingeblendeten Konsole aus und übertitelt sie mit `pT`.

Anfrage **`static String erwarteEingabe()`**
Blendet am unteren Bildschirmrand eine Konsole ein und wartet auf die Eingabe eines String.

Anfrage **`static String erwarteEingabe(String pTitel)`**
Blendet am unteren Bildschirmrand eine Konsole ein und wartet auf die Eingabe eines String. Die Konsole wird mit `pTitel` übertitelt.

Anfrage **`static GLObjekt gibObjekt(double pX, double pY)`**
Gibt das Objekt zurück, welches an der Stelle `(pX,pY)` im Kamerafenster zu sehen ist.

Auftrag **`static void warte()`**
Lässt das System eine Millisekunde warten.

Auftrag **`static void warte(int pM)`**
Lässt das System `pM` Millisekunden warten.