GUI()

Im Konstruktor wird ein neuer Behälter eingefügt. Ich habe mich für ein *JFrame* entschieden, da ich dieses am häufigsten verwende. In den folgenden sechs Methoden definiere ich das Fenster:

* Die erste Methode setzt den Titel des *JFrame*.
* Die zweite Methode beschreibt den Vorgang beim Drücken des „Schliessen“-Knopfes.
* Die dritte Methode setzt die Bildschirmgrösse auf 500 \* 500 Pixel ein.
* *setResizable(false)* verhindert, dass der Benutzer die Fenstergrösse verändern kann.
* Die fünfte Methode zentriert das Fenster sowohl horizontal als auch vertikal.
* Die sechste Methode setzt das Layout auf das gebräuchliche *BorderLayout.*

Die nächste Methode ruft das bereits deklarierte Objekt *cube* auf und instanziiert es als Objekt der Rubiks\_Cube-Klasse.

Danach wird das *cube*-Objekt auf das *JFrame* mit *add()* im Zentrum des Fensters hinzugefügt. Die nächste Methode fügt dem *JFrame* ein *JPanel* am oberen Bildschirmrand hinzu, indem *createPanel()* aufgerufen wird. Zum Schluss wird das *JFrame* für den Endnutzer sichtbar gemacht.

createPanel()

Zu Beginn der Methode werden ein *JPanel*, drei *JButton* und ein *ActionListener* deklariert und instanziiert. Als nächstes wird *actionPerformed(ActionEvent e)* ausprogrammiert. Innerhalb dieser Methode wird überprüft, welcher der drei *JButtons* gedrückt wurde; je nach *JButton* wird eine andere Methode der Klasse Rubiks\_Cube ausgeführt. Zum Schluss werden den *JButtons* der *ActionListener* hinzugefügt und die *JButtons* in das *JPanel* implementiert. Das *JPanel* wird danach zurückgegeben.

(Quelle für drei Programmierworte:

<http://www.informatik-forum.at/showthread.php?40815-Instanzieren-initialisieren-deklarieren>)

Zuerst werden neue Instanzvariablen deklariert. Die fünfte und sechste Variable werden zusätzlich initialisiert. Die letzten drei Variablen werden instanziiert.

|  |
| --- |
| Canvas3D canvas;  TransformGroup objTransform;  String x\_axe;  String y\_axe;  int count = 0;  float angle = 15f;  TransformGroup[][][] stoneTransform = new TransformGroup[3][3][3];  Transform3D[][][] rotate = new Transform3D[3][3][3];  Stone[][][] stone = newStone[3][3][3]; |

Abb. xy: Instanzvaribeln der Rubiks\_Cube Klasse

Rubiks\_Cube()

Im Konstruktor wird die klasseneigene Methoden createViwBranch() aufgerufen und dem *SimpleUniverse* hinzugefügt. Ebenfalls wird dem *SimpleUniverse* durch addBranchGroup(createContentBranch()) die Methode createContentBranch() aufgerufen.

createViewBranchGraph()

Das *Canvas3D* Objekt wird instanziiert und mit SimpleUniverse.getPreferredConfiguration() wird die bevorzugte Grafik-Konfiguration gefunden. Weiter wird das *SimpleUniverse* instanziiert und das *Canvas3D*-Objekt übergeben. Die nächsten zwei Methodenaufrufe setzen die Kameraansicht so, dass zu Beginn der ganze Würfel sichtbar ist.