|  |  |
| --- | --- |
|  | |
|  | Projekt Dokumentation:  Lineare Gleichungssysteme Lösen mittels Gauß-Algorithmus |
|  | Auszubildender: Jacek Langer  Ausbilder: Stefan Tomaszek  1. Ausbildungsjahr **2022 – Q1** |

Inhaltsverzeichnis

[1 Hintergrund 1](#_Toc102029186)

[2 Der Gauß-Algorithmus zum Lösen von Matrizen 1](#_Toc102029187)

[3 Anwendung 2](#_Toc102029188)

[4 Ausnahmen 2](#_Toc102029189)

[5 UML Klassendiagramm 3](#_Toc102029190)

[6 UML Flowcharts 4](#_Toc102029191)

[7 Abbildungsverzeichnis 7](#_Toc102029192)

[8 Literaturverzeichnis 7](#_Toc102029193)

# Hintergrund

Bei LGS handelt es sich um ein Schulprojekt, zum Lösen von Matrizen. Dabei findet der Gauß-Algorithmus Verwendung. Bei der Implementierung handelt es sich um eine Klassenbibliothek die eine Statische Methode bereitstellt. Es wird ein Argument erwartet und ein weiteres ist Optional zu Übergeben. Wird nur ein Argument übergeben wird aus der Übergebenen Matrix die Inverse Matrix gebildet. Wird ein Vector als zweites Argument übergeben wird die Angegebene Matrix anhand des Lösungsvektors gelöst.

# Der Gauß-Algorithmus zum Lösen von Matrizen

Beim Gaußschen Algorithmus zum lösen von Matrizen handelt es sich um ein Iteratives Verfahren zum Lösen von Matrizen anhand eines Lösungsvektors. Dabei wird eine Matrix mit einem Ausgangsvektor gleichgestellt und die einzelnen Zeilen der Matrix mit einander addiert bis die Matrix eine Einheitsmatrix darstellt. Dabei wird auf den Vektor dieselbe Operation angewendet wie auf die Matrix. Bei einer gelösten Matrix können die Werte für die einzelnen Unbekannten aus dem Lösungsvektor abgelesen werden.

(x2+ y2+ z2) + x1 +y1 + z1

Beispiel:

Erste Iteration:

Rückwärtige Iteration:

Normalisieren der Einheitsmatrix

Lösungsvektor:

Abbildung Rechenbeispiel Gaußscher Algorithmus

# Anwendung

Der Statischen Methode SolveLgs der Klasse Gauss werden zunächst eine Matrix und optional ein Vektor übergeben. Wird der Vektor nicht angegeben wird die Inverse der Matrix gebildet. Wurde eine Vektor angegeben wird die Matrix gelöst sofern möglich und ein Lösungsvektor ausgegeben. Sollte eine Matrix angegeben worden sein die nicht Quadratisch ist (z.B. 4x3), wird eine Ausnahme geworfen.

Beispiel:

var actual = Gauss.SolveLgs(matrix, resultVector);

# Ausnahmen

Sollte die Matrix nicht lösbar sein wird eine Ausnahme geworfen. Dabei handelt es sich um eine Ausnahme vom Typ: UnsolvableMatrixException.

var resultVector = new Vector(new double[] {5, 6});  
var matrix = new Matrix(new[]  
{  
 new double[] {2, 4},  
 new double[] {2, 4}  
});

Gauss.SolveLgs(matrix, resultVector);

----------------------------------------------------------------------------------

Console output:

----------------------------------------------------------------------------------

LF5\_LGS.exceptions.UnsolvableMatrixException : this matrix cannot be solved with a LGS

Abbildung Ausnahme UnsolvableMatrixException

Sollten eine Matrix und ein Lösungsvektor übergeben werden, welche unterschiedliche Größen aufweisen, wird eine IllegalMatrixOperationException geworfen.

# UML Klassendiagramm

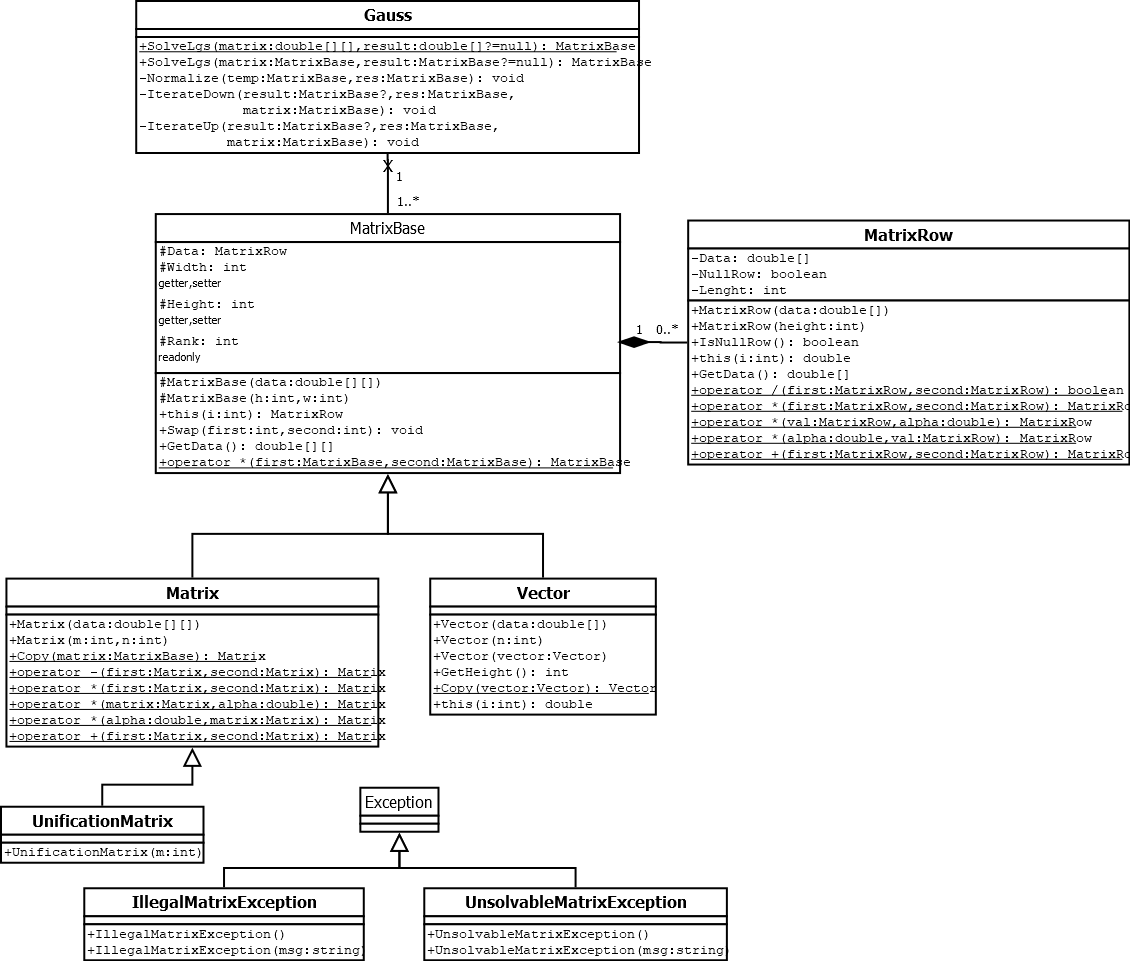


Abbildung UML Klassen Diagramm

# UML Flowcharts

Abbildung SolveLgs Methode

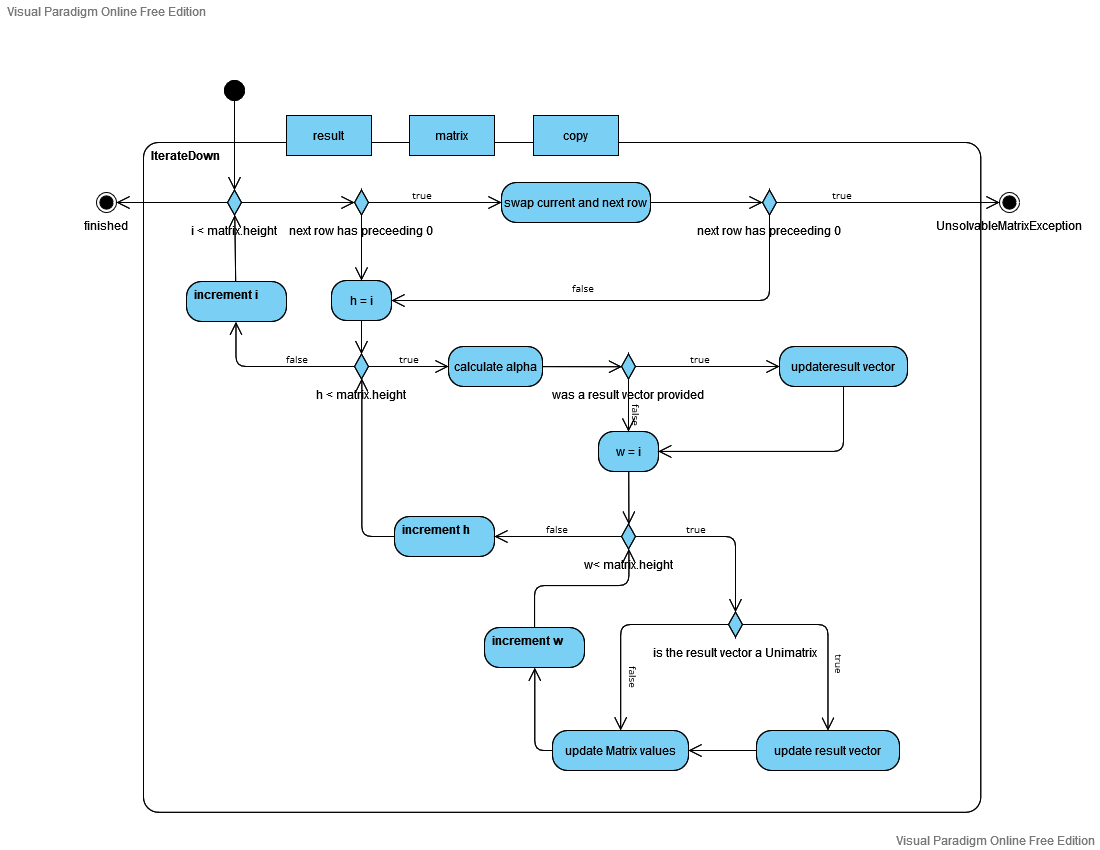
Abbildung Iterate Down Methode

Abbildung 6 Iterate Up Methode

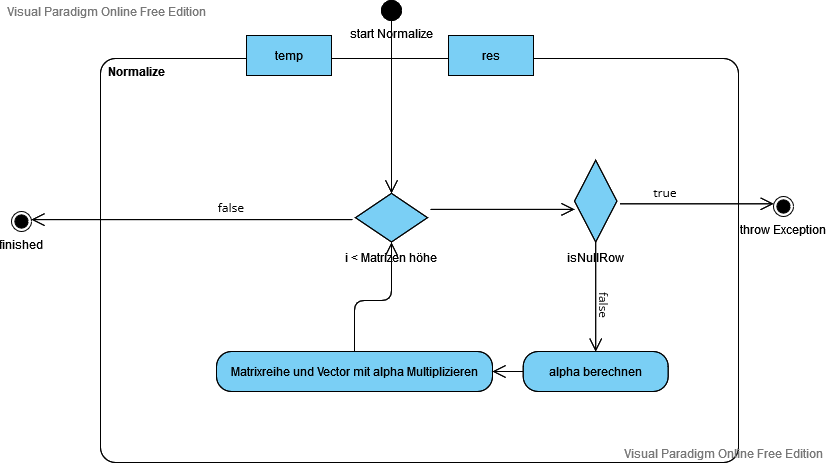


Abbildung Normalize Methode

# Abbildungsverzeichnis

[Abbildung 1 Rechenbeispiel Gaußscher Algorithmus 1](#_Toc102029194)

[Abbildung 2 Ausnahme UnsolvableMatrixException 2](#_Toc102029195)

[Abbildung 3 UML Klassen Diagramm 3](#_Toc102029196)

[Abbildung 4 SolveLgs Methode 4](file:///C:\Users\langer\Desktop\OSZIMT\LF5\c%23\LF5_LGS\Dokumentation.docx#_Toc102029197)

[Abbildung 5 Iterate Down Methode 5](#_Toc102029198)

[Abbildung 6 Iterate Up Methode 6](#_Toc102029199)

[Abbildung 7 Normalize Methode 7](#_Toc102029200)

# Literaturverzeichnis