

# Bericht - Übung 1

**Gruppe:** Mineral - Lena Meßmer, Sophie Wirth, Christian Mehns

## Bilder

### Programm zum Öffnen und Anzeigen von Bilder

ImageLoader erweitert JFrame.

Im Konstruktor befindet sich der gesamte Code.

Zunächst wird ein *JFileChooser* Objekt erzeugt, um über einen Dialog die Datei-Auswahl des Benutzers abzufragen. Um die Auswahl dabei auf Bilder vom Dateityp jpg, jpeg oder png einzuschränken, wird ein *FileNameExtensionFilter* erzeugt und dem FileChooser hinzugefügt. Damit der Benutzer nur ein Bild gleichzeitig auswählen kann, wird *setMultiSelectionEnabled* auf false gesetzt.

Der Rückgabewert (int) der Methode *showOpenDialog* wird benutzt, um zu überprüfen ob eine Datei ausgewählt worden ist. Wenn dies der Fall ist, so wird der Pfad der ausgewählten Datei in der *File*-Variablen "file" gespeichert. Diese wird nun benutzt, um den Titel des Fensters auf den des Bild-Titels zu setzen und die Bilddatei in ein *BufferedImage* zu lesen. Da es beim Einlesen von Dateien zu einer *IOException* kommen kann, wird die *read*-Methode in einem try- catch- Block aufgerufen.

### Programm das ein Bild erzeugt mit rotem Strich auf schwarzem Hintergrund

ImageSaver implementiert ActionListener

Im Konstruktor wird ein Fenster mit den Maßen 640x480 erzeugt.

Desweiteren wird eine Menüleiste erzeugt mit einem Auswahlelement, File. File hat ein Untermenü, Save, womit man das erzeugte Bild abspeichern können soll. Dann wird das in

ImageCanvas erzeugte schwarze Canvas mit roter Diagonale in das Fenster eingefügt und sichtbar gemacht.

In der Methode actionPerformed wird ein Speicherdialog geöffnet um den Benutzer das erzeugte Bild speichern zu lassen. Zuerst wird geprüft ob Save gewählt wurde. Es wird eine JFileChooser Variable 'fileDialog' kreiert, deren Rückgabewert in der int Variable 'returnValue' gespeichert wird dazu genutzt um festzustellen ob eine Datei zum Speichern ausgewählt wurde. Dann wird die ausgewählte Datei in einer Variable 'saveImage' gespeichert, um den Namen zu bekommen und entscheiden zu können ob die Datei in ".jpg" oder ".png" endet, wenn das nicht der Fall ist muss eine Fehlermeldung geworfen werden, die Aufschluss darüber gibt das das Bild eine dieser Endungen haben muss. Danach wird in einem try Block eine neue Datei in das System geschrieben, mit der ImageIO.write Methode die als Parameter die Variablen der ImageCanvas, saveFile und savingImage bekommt. Exception werden im catch Block abgefangen und ein StackTrace wird auf der Konsole ausgedruckt.

In der Main Methode wird der Konstruktor aufgerufen.

## **Matrizen- und Vektorenbibliothek**

### **Mat3x3**

Instanzen dieser Klasse repräsentieren eine 3x3 Matrix mit neun Einträgen/Komponenten. Dabei wird für jede Instanz die Determinante der jeweiligen Matrix zur Verfügung gestellt.

Zusätzlich sind über Methoden folgende Matrixoperationen anwendbar:

Matrix-Matrix-Produkt (Mat3x3 mul (final Mat3x3 m)),

Matrix-Vektor-Produkt (Vector3 mul (final Vector3 m)),

Matrix \* Punkt (Point3 mul (final Point3 p))

sowie das Tauschen einzelner Spalten der Matrix gegen einen gegebenen Vektor (changeCol1-3).

Die Operationen wurden nach den jeweiligen mathematischen Vorgaben implementiert. Pro Operation existiert eine Methode.

### **Vector3**

Instanzen dieser Klasse repräsentieren einen Vektor mit drei Komponenten.

Dabei wird für jeden Vektor auch seine Länge bereitgestellt (magnitude).

Zusätzlich sind über Methoden folgende Operationen anwendbar:

Vektoraddition (Vektor+Vektor und Vektor+Normale),

Subtraktion Vektor-Normale,  
Multiplikation Vektor und reelle Zahl,  
Skalarprodukt (Vektor mit Vektor und Vektor mit Normale),  
Reflexion des Vektors an einer Normalen,  
Kreuzprodukt zweier Vektoren  
sowie die Möglichkeit den Vektor zu normalisieren oder in eine Normale zu schreiben.

Die Operationen wurden nach den jeweiligen mathematischen Vorgaben implementiert.  
Pro Operation existiert mindestens eine Methode (bei Varianten sind es mehrere Methoden).

### **Point3**

Instanzen dieser Klasse repräsentieren einen Punkt im dreidimensionalen Raum.

Es ist möglich Vektoren zu/von einem Punkt zu addieren/subtrahieren.

### **Normal3**

Instanzen dieser Klasse repräsentieren eine Normale mit drei Komponenten.

Zusätzlich sind über Methoden folgende Operationen anwendbar:

Multiplikation Normale und reelle Zahl,  
Normalenaddition,  
Skalarprodukt (Normale mit Vektor),

Die Operationen wurden nach den jeweiligen mathematischen Vorgaben implementiert.  
Pro Operation existiert eine Methode.