

## **Projekthinhalt:**

Wir entwickeln eine Java-basierte Desktop App zur Videoverarbeitung, die mithilfe von Java Swing realisiert wird. Das Hauptziel des Projekts ist es, Videodateien aus einem definierten Input-Verzeichnis zu lesen, diese rekursiv zu verarbeiten und in Segmente aufzuteilen. Dabei wird ein Wasserzeichen eingefügt, das mithilfe einer „geheimen“ Tastenkombination Strg+Alt+S in einem zusätzlichen Fenster konfiguriert werden kann. Weitere Funktionen umfassen das Speichern und Laden von Einstellungen (unter Verwendung von JSON als Dateiformat) sowie das Logging sämtlicher Aktivitäten in einem separaten Ordner namens „logs“. Neben den Mindestanforderungen wie Klassenstruktur, Fehlerabfang, GUI mit zwei Fenstern, Dialogfenstern und Grafik (zum Beispiel ein angezeigtes Bild in der Haupt-GUI), wurden zusätzliche Anforderungen wie Rekursion, die Einbindung einer externen Bibliothek (org.json) sowie die Verwendung eines Enums zur Darstellung unterschiedlicher Verarbeitungsmodi umgesetzt.

## **Aufgabenverteilung:**

### **1. Nikola:**

- Übernahme der GUI-Entwicklung (Paket „app“), einschließlich der Umsetzung der Hauptfenster (MainWindow) und der zusätzlichen Fenster (SettingsWindow und WatermarkWindow).
- Implementierung der geheimen Tastenkombination (Ctrl+Alt+S) zur Konfiguration des Wasserzeichens.
- Gestaltung der Dialogfenster und Einbindung von grafischen Elementen (z.B. Bilder).

### **2. Patrick:**

- Entwicklung der Backend-Logik zur Videoverarbeitung (Paket „utils“), einschließlich der rekursiven Dateisuche, fehlerfreundlichen Bedienung sowie der Logik zum Einfügen bzw. Entfernen eines Wasserzeichens.
- Implementierung der Datenspeicherung und -verwaltung über JSON (Settings) und Logging in den Ordner „logs“.
- Erstellung von Unit Tests (im Paket „test“) zur Absicherung der Funktionalität.

## **Umsetzung der Anforderungen:**

### **1. Klassen und Packages:**

- Es wurden mindestens vier Klassen implementiert (MainApp, MainWindow, SettingsWindow, WatermarkWindow im GUI-Bereich sowie VideoProcessor, Settings und Logger im Logik-Bereich).
- Es kommen zwei Hauptpackages zum Einsatz: „app“ (für GUI) und „utils“ (für Verarbeitung, Logging und Einstellungen).
- Zusätzlich wurde eine Enum-Klasse (ProcessorType) eingeführt.

### **2. Vererbung:**

- Die GUI-Klassen nutzen teilweise Vererbungsstrukturen, indem etwa gemeinsame Funktionalitäten in Basisklassen oder Utility-Klassen ausgelagert werden.

### **3. Datenspeicherung:**

- Einstellungen werden in einer JSON-Datei im Ordner „config“ gespeichert und beim Programmstart wieder geladen.
- Log-Nachrichten werden systematisch in den Ordner „logs“ geschrieben (dazu wird beim Programmstart sichergestellt, dass dieser Ordner existiert; falls nicht, wird er automatisch erzeugt).

### **4. Fehlerabfang:**

- Im gesamten Projekt wird Exception Handling verwendet, sodass Fehler benutzerfreundlich über Dialogfenster oder Logfiles ausgegeben werden.

### **5. Benutzereingabe/-Interaktion & GUI:**

- Die Haupt-GUI bietet Buttons für den Start der Videoverarbeitung, den Aufruf der Einstellungen und die Anzeige des aktuellen Status.

- Zwei separate GUI-Fenster (SettingsWindow und WatermarkWindow) gewährleisten eine klare Trennung der Funktionen.

- Zusätzlich werden Dialogfenster (z. B. JOptionPane) verwendet, um dem Benutzer Feedback zu Aktionen (Speichern der Einstellungen, Fehlermeldungen etc.) zu geben.

## **6. Graphics:**

- In der Haupt-GUI werden grafische Elemente (zum Beispiel Bilder) eingebunden, um das Interface ansprechend zu gestalten.

## **7. Zusätzliche Anforderungen:**

Rekursion: Die Video-Suchfunktion scannt untergeordnete Ordner rekursiv nach Videodateien.

Einbindung/Verwendung externer Libraries: Die JSON-Verarbeitung erfolgt mithilfe der externen Bibliothek org.json.

Enum: Die Verarbeitung unterschiedlicher Modi wird unter anderem durch ein Enum (ProcessorType) repräsentiert.

Unit Tests: Es sind Unit Tests für wichtige Komponenten (z.B. Settings) im Paket „test“ implementiert.

## **Zusätzliche Features:**

Eine geheime Tastenkombination Strg+Alt+S wurde implementiert, die das dev Fenster öffnet, in dem das Wasserzeichen konfiguriert werden kann. So wird das Wasserzeichen bei Videos eingeblendet, außer im Falle einer aktivierten Pro-Version, bei der es automatisch deaktiviert wird. (Es gibt keine Pro version)

Das Logging-System sorgt dafür, dass alle wichtigen Aktionen in einem eigenen Log-Ordner protokolliert werden.

Dieses Konzept bietet eine strukturierte und professionelle Umsetzung der vorgegebenen Anforderungen und stellt sicher, dass alle relevanten Aspekte (Verwaltung von Einstellungen, Grafik, Fehlerbehandlung, rekursive Verarbeitung, Logging, etc.) abgedeckt sind.