**Zusammenfassung: MaehrDocs Modularisierung**

**Projekt/Problemstellung**

Wir haben ein autodocs\_v2.py-Programm für die automatische Verarbeitung von PDF-Dokumenten mit einer GUI namens "MaehrDocs" modularisiert. Das ursprüngliche Programm analysiert PDFs mit Hilfe von OpenAI's KI, extrahiert relevante Informationen und benennt die Dateien nach einem Schema um. Es erkennt Duplikate, behandelt Fehler und sortiert Dokumente in entsprechende Ordner. Durch die Modularisierung wurde der Code in logische Komponenten aufgeteilt, um die Wartbarkeit zu verbessern.

**Wichtige Befehle und Eingaben**

1. **Paketstruktur erstellen:**
   * Erstellung eines Hauptordners für das Projekt
   * Erstellung einer \_\_init\_\_.py im Hauptordner
   * Erstellung eines Unterordners maehrdocs als Hauptpaket
   * Erstellung einer \_\_init\_\_.py im maehrdocs-Ordner
2. **Module erstellen:**
   * config.py - Konfigurationsverwaltung mit ConfigManager-Klasse
   * document\_processor.py - Dokumentenverarbeitung mit DocumentProcessor-Klasse
   * duplicate\_detector.py - Überarbeitet mit DuplicateDetector-Klasse
   * gui\_manager.py - GUI-Verwaltung mit GuiManager-Klasse
3. **Hauptdateien:**
   * main.py - Haupteinstiegspunkt mit Kommandozeilenargumenten
   * start\_maehrdocs.py - Vereinfachter Einstiegspunkt für die GUI
   * maehrdocs.bat - Batchdatei zum einfachen Starten der GUI
4. **Paketexport einrichten:**
   * Aktualisierung der \_\_init\_\_.py im Hauptmodul mit Imports und Version
   * Erstellung einer README.md mit Projektinformationen

**Erkenntnisse / Entscheidungen**

1. **Modulare Struktur:**
   * Trennung von Konfiguration, Dokumentverarbeitung, Duplikatserkennung und GUI
   * Jede Komponente als eigenständige Klasse in separaten Dateien
2. **Verbesserte Importstruktur:**
   * Klare Import-Hierarchie zwischen den Modulen
   * Zentraler Export der wichtigsten Komponenten in \_\_init\_\_.py
3. **Mehrere Einstiegspunkte:**
   * Kommandozeilen-Interface in main.py für vollständige Kontrolle
   * Vereinfachter GUI-Start über start\_maehrdocs.py
   * Benutzerfreundlicher Start über maehrdocs.bat
4. **Optimierte Batch-Datei:**
   * Vereinfachte Version mit Titelausgabe und Feedback
   * Entfernung überflüssiger Prüfungen, die jetzt in Python-Modulen behandelt werden

**Offene Punkte / Nächster Schritt**

1. **Testen der modularisierten Struktur:**
   * Überprüfung, ob alle Module korrekt zusammenarbeiten
   * Testen der verschiedenen Startmethoden (GUI, Kommandozeile)
2. **Überprüfung der Import-Pfade:**
   * Sicherstellen, dass alle relativen Imports korrekt funktionieren
3. **Dokumentation vervollständigen:**
   * Docstrings zu allen Klassen und Methoden hinzufügen
   * Kommentare für komplexe Codeabschnitte ergänzen
4. **Übergangsplan:**
   * Alte Dateien entfernen oder umbenennen, sobald die neue Struktur stabil läuft

**Stand: Mai 2025**

Die Modularisierung des MaehrDocs-Programms wurde erfolgreich durchgeführt. Alle wesentlichen Komponenten wurden in separate Module mit klaren Zuständigkeiten aufgeteilt. Die Hauptfunktionalität wurde in einer ConfigManager-Klasse, einer DocumentProcessor-Klasse, einer DuplicateDetector-Klasse und einer GuiManager-Klasse gekapselt.

Es wurden neue Einstiegspunkte erstellt: eine main.py-Datei für Kommandozeilenoptionen und eine start\_maehrdocs.py-Datei für den einfachen GUI-Start. Die maehrdocs.bat-Datei wurde aktualisiert, um die neue Struktur zu unterstützen.

Als nächster Schritt sollte die neue modulare Struktur getestet werden, um sicherzustellen, dass alle Komponenten korrekt zusammenarbeiten und dass die Funktionalität des ursprünglichen Programms vollständig erhalten bleibt.