**htl donaustadt**
Donaustadtstraße 45
1220 Wien
www.htl-donaustadt.at

htl donaustadt

# SEW-Test – 3AHITN 08.05.25

| **Name:** | **Punkte:** |
|---|---|
| | **Note:** |

Entwickeln Sie eine GUI-Anwendung, um eine ToDo-Liste zu verwalten. Ergänzen Sie dazu den bestehenden Code. Es gibt 2 Fenster, das Hauptfenster ToDoApp, wo die Hauptlogik durchgeführt wird. ToDoDialog ist ein zweites Fenster um To-Do-List Items hinzuzufügen oder zu ändern. Alle Einfügungen, Änderungen und Löschungen an der internen ArrayList<TodoItem> sollen direkt im ToDoDialog geschehen.

## JSON Klasse „ToDoJsonManager“:

**1. JSON-Klasse anpassen:**

Passen Sie ToDoJsonManager so an, dass statt einzelner Objekte jetzt eine ArrayList<TodoItem> geladen und gespeichert wird.

## 2. Hauptaufgabe

1. **Dialog-Integration**
   - Öffnen Sie den ToDoDialog bei „Neu“ und „Bearbeiten“, übergeben Sie dort die ArrayList<TodoItem> und ggf. den Index.
   - Im Dialog fügen Sie bei „Neu“ ein Element in die Liste ein, bzw. ersetzen bei „Bearbeiten“ das bestehende.
   - Wichtig: Bevor der Dialog sich schließt (dispose())
   - Hinweis: rufen Sie am besten public-Methode in ToDoApp auf die die TextArea mit den geänderten/neuen Items anzeigt.
2. **Anzeigen im Hauptfenster**
   - Im Hauptfenster werden alle To-Dos mit Nummer, Titel, Erledigt-Status, Priorität und Kategorie zeilenweise in einer TextArea dargestellt.
   - Ein Eingabefeld nimmt die Nummer (Index) einer To-Do-Zeile auf, um sie zu bearbeiten oder löschen.
3. **Einfache Fehlerüberprüfung**
   - Stellen Sie sicher, dass der eingegebene Index eine gültige Zahl ist.
   - Überprüfen Sie dass die Felder für Titel, Prioritär, etc. nicht leer sind (Beschreibung ist nicht unbedingt notwendig).
   - Bei ungültigen Eingaben zeigen Sie eine Fehlermeldung mit JOptionPane und blockieren das Weitermachen.
4. **Laden & Speichern**
   - Beim Klick auf „Laden“ rufen Sie loadToDos() in einem try-catch auf und zeigen im Fehlerfall eine JOptionPane-Meldung.
   - Analog für „Speichern“ und saveToDos(...).

## 3. Zusatzaufgaben

- **ProgressBar**
  Fügen Sie im Hauptfenster eine JProgressBar hinzu, die den Anteil der bereits erledigten To-Dos anzeigt.
- **Erweiterte Eingabe-Validierung & Exception-Handling**
  Die Methoden loadToDos() und saveToDos(List<TodoItem>) sollen throws IOException deklarieren und nicht mehr intern abfangen.
  Validieren Sie alle Eingaben mit String-Manipulationen (z. B. trim(), matches("\\d+")) und fangen Sie NumberFormatException sowie IOException ab.
- **Farbwahl**
  Entkommentieren Sie das Attribut Color farbe in TodoItem. In ToDoDialog bieten Sie einen „Farbe wählen…“-Button an, der mit JColorChooser arbeitet. Speichern Sie die Farbe im Objekt und verwenden Sie sie ggf. bei der Anzeige.
- **Sortierung nach Priorität**
  Lassen Sie die Liste vor der Anzeige so sortieren, dass Elemente mit Priorität 5 zuerst stehen.

# Arbeiten Sie in C:/Arbeitsordner und geben Sie Ihr Projekt am Schluss im Z: Laufwerk ab!