



Hochschule Macromedia für angewandte Wissenschaften
University of Applied Sciences

Kursbezeichnung: Aufbaumodul Programmieren

Prüfer: Herr Thomas Berger

Vom Studierenden auszufüllen:

305136

Matrikelnummer

D-PBf DT DFO 6d 24W

Kohorte

Ramis

Marie Grace

Nachname

Vorname

Die Arbeit wird eingereicht als:

(Tragen Sie bitte in die zutreffende Box den Buchstaben X ein)

☒

Einzelarbeit

☐

Gruppenarbeit

Trifft nur auf Gruppenarbeiten zu: (Nur bei Gruppenarbeiten auszufüllen)

Falls Sie eine Gruppenarbeit einreichen, dann müssen bitte die Vor- und Nachnamen aller Gruppenmitglieder aufgeführt werden. Die Namen sind von den jeweiligen Gruppenmitgliedern selbst elektronisch einzutragen. Durch Eintrag des Namens wird bestätigt, dass der/die Studierende mit der Abgabe der Arbeit in der vorliegenden Form einverstanden ist. Ferner wird mit Eintrag des Namens erklärt, die Projektarbeit (bei Gruppenarbeit: den von dem/der jeweiligen Studierenden erstellten und entsprechend gekennzeichneten Teil der Arbeit selbständig und ohne fremde Hilfe angefertigt zu haben. Dabei hat der/die Studierende sich keiner anderen Hilfsmittel bedient als derjenigen, die im beigelegten Quellenverzeichnis genannt sind. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind von dem/der Studierenden als solche kenntlich gemacht. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

- 1)
2)
3)
4)

- 5)
6)
7)
8)

Bewertung der Gruppenarbeiten:

(Tragen Sie bitte in die zutreffende Box den Buchstaben X ein)

☐

Ich beantrage bei meiner Gruppenarbeit eine Individualbewertung

☐

Ich beantrage bei meiner Gruppenarbeit eine Gruppenbewertung

Eine Individualbewertung bedeutet, dass jedes Gruppenmitglied eine individuelle Note bekommt und eine Gruppenbewertung bedeutet, dass jedes Gruppenmitglied eine identische Note bekommt.

Overath, 20.05.2025

Ort/Datum

Marie Grace Ramis

Vollständiger Vor- und Nachname

Vom Prüfer auszufüllen: (Freitext für die Zweitkorrektur)

Hochschule Macromedia für angewandte Wissenschaften,
University of Applied Sciences

Aufbaumodul Programmieren

zur Erlangung des akademischen Grades

Bachelor of Arts

To Do App

im Studiengang Digital Technologies and Coding

Studienrichtung Digital Forensics

Erstprüfer:

Herr Thomas Berger

Vorname Name: Marie Grace Ramis

Matr.-Nr.: 305136

Studiengang: Digital Technologies and Coding

Fachrichtung: Digital Forensics

Düsseldorf, im Mai, 2025

Inhalt

1. To Do List.....	1
1.1 Objektorientierte Entwicklung	1
1.2 Vererbung	3
1.3 Pakete/ Module.....	3
2. KI Verzeichnis	5
3. Eidesstattliche Erklärung	6

1. To Do List

Das erstellte Programm ist eine To Do App, die von jedem Nutzer verwendet werden kann. Die App wurde mit angenehmen Farben und einer ansprechender Schrift versehen. Sie wurde bewusst simple aufgebaut, um die Nutzung der App so einfach zu gestalten wie möglich.

Das Programm wurde in drei verschiedenen Klassen aufgeteilt (Task, User und ToDoApp). Diese Trennung ermöglicht eine saubere Strukturierung der verschiedenen Aufgabenbereiche: Aufgabenverwaltung, nutzerbezogene Daten und grafische Benutzeroberfläche (GUI). Dazu bietet die App die grundlegenden Funktionen mit einem Login/ Logout, Aufgaben hinzufügen, Aufgaben als erledigt markieren und Erinnerungen setzen.

Die Klasse ToDoApp erbt von tk.Tk, wodurch die grafische Oberfläche mit dem tkinter-Framework realisiert wurde. Zusätzlich wurden die Bibliotheken json und os verwendet. Eine Modularisierung ist durch die Aufteilung in Klassen und Methoden erkennbar.

Die App mit den Basisfunktionen ist einfach gestaltet, damit die Nutzer sich selbstständig gut durch die App navigieren können. Ziel dieses Programm ist es, eine einfache und übersichtliche To Do Liste zur Verfügung zu stellen, mit der alltägliche Aufgaben organisiert und verwaltet werden können. Die To Do App ermöglicht das Erstellen, Bearbeiten und Löschen von Aufgaben sowie das Markieren wichtiger Aufgaben und Erinnerungen. Für die Funktionalität wird eine Benutzerfreundliche Oberfläche und eine klare Struktur genutzt, die es ermöglicht, jeden Nutzer im Alltag zu unterstützen,

1.1 Objektorientierte Entwicklung

Für die objektorientierte Programmierung (OOP) wurden jeweils Klassen erstellt, die unterteilt in Task, User und ToDoApp sind. Die Klasse Task enthält Eigenschaften und Funktionen, die eine einzelne Aufgabe beschreiben, zum Beispiel:

```
class Task:
    """Representing the various functions like completion, reminding, and marking the
    task as important"""
    def __init__(self, description):
        """adding descriptions to each task"""
```

```
self.description = description
self.completed = False
self.important = False
self.reminder = None

def mark_completed(self):
    """task completed"""
    self.completed = True
```

Die Klasse User wurde programmiert, um die Nutzerinformationen mit den zugehörigen Aufgaben zu verbinden wie z.B.:

```
class User:
    """Showing the user with the chosen username, password, and the list of tasks they
    added"""
    def __init__(self, username, password):
        self.username = username
        self.password = password
        self.tasks = []

    def add_task(self, description):
        """adding task"""
        task = Task(description)
        self.tasks.append(task)
```

Die letzten Klasse ToDoApp bildet die eigentliche To Do Liste ab und wurden mit der GUI umgesetzt. In dieser Klasse werden alle Funktionen der App visuell dargestellt und bedienbar gemacht, zum Beispiel:

- Darstellung der Aufgabenliste
- Buttons zum Hinzufügen, Bearbeiten oder Löschen von Aufgaben
- Login- und Logout-Funktion
- Erinnerungen hinzufügen und anzeigen

Beispiel aus dem Programm:

```

class ToDoApp(tk.Tk):
    """Main application class/ includes the GUI as well as the user interactions"""
    def __init__(self):
        """Initialize the to-do list app GUI and loads the login screen"""
        super().__init__()
        self.title("To-Do List App")
        self.geometry('400x500')
        self.configure(bg='#f0f8ff')
        self.users = {}
        self.load_data() # Load data from json file
        self.current_user = None
        self.login_screen()

```

1.2 Vererbung

Die Vererbung wird in der Klasse ToDoApp genutzt, welche von tk.Tk erbt. Dadurch bekommt die Klasse Zugriff auf sämtliche GUI Funktionen des tkinter Modul, was die Erstellung und Steuerung der GUI ermöglicht:

```

class ToDoApp(tk.Tk):
    """Main application class/ includes the GUI as well as the user interactions"""
    def __init__(self):
        """Initialize the to-do list app GUI and loads the login screen"""
        super().__init__()
if __name__ == '__main__':
    app = ToDoApp()
    app.mainloop()

```

1.3 Pakete/ Module

Module und Packte, die verwendet werden, werden direkt am Anfang des Codes importiert:

```

#importin GUI
import tkinter as tk
from tkinter import messagebox, simpledialog

```

```
#importing json
import json
import os
```

In diesem Programm werden verschiedene Standardbibliotheken verwendet. Tkinter dient dazu, die GUI darzustellen, über die die Nutzer mit der App interagieren können. Json wird verwendet, um die Daten von allen Nutzer zu speichern. Dadurch ist es möglich, die Login Daten und To Do Liste immer wieder aufzurufen. Das os Modul wird für Datei Handling verwendet z.B. um zu prüfen ob, bestimmte Dateien bereits vorhanden sind.

2. KI Verzeichnis

Prompt 1

Zeile: 61-72

Wortlaut des Prompts: Korrigiere damit die Datei von json lädt.

Übernahme sowie Erklärung der Herangehensweise: Die Korrektur wurde zur teils übernommen.

Prompt 2

Zeile: 74- 88

Wortlaut des Prompts: Korrigiere (einsetzen des Codes)

Übernahme sowie Erklärung der Herangehensweise: Die Korrektur wurde ganz übernommen.

Prompt 3

Zeile: 114- 128

Wortlaut des Prompts: Erkläre warum die Buttons werden nicht angezeigt.

Übernahme sowie Erklärung der Herangehensweise: Die Korrektur wurde ganz übernommen.

3. Eidesstattliche Erklärung

Ich, Marie Grace Ramis

geboren am 02.08.1997

erkläre hiermit, die vorliegende Arbeit
selbständig und ohne fremde Hilfe
angefertigt zu haben. Dabei habe ich
mich keiner anderen Hilfsmittel bedient
als derjenigen, die im beigefügten
Quellenverzeichnis genannt sind.

Alle Stellen, die wörtlich oder sinngemäß
aus Veröffentlichungen entnommen
wurden, sind von mir als solche
kenntlich gemacht.

Düsseldorf, 20.05.2025

Marie Grace Ramis

....., den

.....

Studienort

Unterschrift Studierende/r (= Verfasser/in)