import pickle

import tkinter as tk

from tkinter import messagebox, simpledialog

# Task class to represent a to-do item

class Task:

def \_\_init\_\_(self, description, completed=False):

self.description = description

self.completed = completed

def \_\_str\_\_(self):

return f"{self.description} [{'✓' if self.completed else '✗'}]"

# ToDoList class to manage tasks

class ToDoList:

def \_\_init\_\_(self, filename="tasks.pkl"):

self.filename = filename

self.tasks = self.load\_tasks()

def add\_task(self, description):

self.tasks.append(Task(description))

def list\_tasks(self):

return [str(task) for task in self.tasks]

def mark\_task\_completed(self, index):

if 0 <= index < len(self.tasks):

self.tasks[index].completed = True

def delete\_task(self, index):

if 0 <= index < len(self.tasks):

del self.tasks[index]

def save\_tasks(self):

with open(self.filename, 'wb') as file:

pickle.dump(self.tasks, file)

def load\_tasks(self):

try:

with open(self.filename, 'rb') as file:

return pickle.load(file)

except (FileNotFoundError, EOFError):

return []

# CLI interface

def cli\_interface(todo\_list):

while True:

print("\n1. Add Task")

print("2. List Tasks")

print("3. Mark Task as Completed")

print("4. Delete Task")

print("5. Save and Exit")

choice = input("Choose an option: ")

if choice == '1':

description = input("Enter task description: ")

todo\_list.add\_task(description)

elif choice == '2':

tasks = todo\_list.list\_tasks()

for i, task in enumerate(tasks, 1):

print(f"{i}. {task}")

elif choice == '3':

tasks = todo\_list.list\_tasks()

for i, task in enumerate(tasks, 1):

print(f"{i}. {task}")

index = int(input("Enter task number to mark as completed: ")) - 1

todo\_list.mark\_task\_completed(index)

elif choice == '4':

tasks = todo\_list.list\_tasks()

for i, task in enumerate(tasks, 1):

print(f"{i}. {task}")

index = int(input("Enter task number to delete: ")) - 1

todo\_list.delete\_task(index)

elif choice == '5':

todo\_list.save\_tasks()

break

else:

print("Invalid choice. Please try again.")

# GUI interface

class ToDoApp:

def \_\_init\_\_(self, root, todo\_list):

self.root = root

self.root.title("To-Do List")

self.todo\_list = todo\_list

# GUI components

self.task\_entry = tk.Entry(root, width=40)

self.task\_entry.pack(pady=10)

self.add\_button = tk.Button(root, text="Add Task", command=self.add\_task)

self.add\_button.pack(pady=5)

self.task\_listbox = tk.Listbox(root, width=50, height=10)

self.task\_listbox.pack(pady=10)

self.mark\_button = tk.Button(root, text="Mark Completed", command=self.mark\_task\_completed)

self.mark\_button.pack(pady=5)

self.delete\_button = tk.Button(root, text="Delete Task", command=self.delete\_task)

self.delete\_button.pack(pady=5)

self.load\_tasks()

def add\_task(self):

description = self.task\_entry.get()

if description:

self.todo\_list.add\_task(description)

self.update\_task\_listbox()

self.task\_entry.delete(0, tk.END)

else:

messagebox.showwarning("Warning", "Task description cannot be empty!")

def mark\_task\_completed(self):

selected\_index = self.task\_listbox.curselection()

if selected\_index:

index = selected\_index[0]

self.todo\_list.mark\_task\_completed(index)

self.update\_task\_listbox()

def delete\_task(self):

selected\_index = self.task\_listbox.curselection()

if selected\_index:

index = selected\_index[0]

self.todo\_list.delete\_task(index)

self.update\_task\_listbox()

def update\_task\_listbox(self):

self.task\_listbox.delete(0, tk.END)

tasks = self.todo\_list.list\_tasks()

for task in tasks:

self.task\_listbox.insert(tk.END, task)

def load\_tasks(self):

tasks = self.todo\_list.list\_tasks()

for task in tasks:

self.task\_listbox.insert(tk.END, task)

# Main function to choose between CLI and GUI

def main():

todo\_list = ToDoList()

choice = simpledialog.askstring("Select Interface", "Enter 'cli' for Command Line or 'gui' for Graphical Interface:").strip().lower()

if choice == 'cli':

cli\_interface(todo\_list)

elif choice == 'gui':

root = tk.Tk()

app = ToDoApp(root, todo\_list)

root.mainloop()

else:

print("Invalid choice. Please enter 'cli' or 'gui'.")

if \_\_name\_\_ == "\_\_main\_\_":

main()