import csv

class Employee:

def \_\_init\_\_(self, name, age, salary):

self.\_\_name = name

self.\_\_age = age

self.\_\_salary = salary

def get\_name(self):

return self.\_\_name

def set\_name(self, name):

self.\_\_name = name

def get\_age(self):

return self.\_\_age

def set\_age(self, age):

self.\_\_age = age

def get\_salary(self):

return self.\_\_salary

def set\_salary(self, salary):

self.\_\_salary = salary

class Manager(Employee):

def \_\_init\_\_(self, name, age, salary, department):

super().\_\_init\_\_(name, age, salary)

self.\_\_department = department

def get\_department(self):

return self.\_\_department

def set\_department(self, department):

self.\_\_department = department

class Worker(Employee):

def \_\_init\_\_(self, name, age, salary, hours\_worked):

super().\_\_init\_\_(name, age, salary)

self.\_\_hours\_worked = hours\_worked

def get\_hours\_worked(self):

return self.\_\_hours\_worked

def set\_hours\_worked(self, hours\_worked):

self.\_\_hours\_worked = hours\_worked

class EmployeeManagementSystem:

def \_\_init\_\_(self):

self.employees = []

def add\_employee(self, employee):

self.employees.append(employee)

self.\_save\_to\_csv(employee)

def display\_employees(self):

for employee in self.employees:

print(employee.\_\_dict\_\_)

def update\_employee(self, name, attribute, value):

for employee in self.employees:

if employee.get\_name() == name:

setattr(employee, f"set\_{attribute}", value)

self.\_update\_csv(employee)

def delete\_employee(self, name):

self.employees = [employee for employee in self.employees if employee.get\_name() != name]

self.\_save\_all\_to\_csv()

def \_save\_to\_csv(self, employee):

with open('employees.csv', 'a', newline='') as file:

writer = csv.writer(file)

writer.writerow([employee.get\_name(), employee.get\_age(), employee.get\_salary(),

getattr(employee, 'get\_department', '')(), getattr(employee, 'get\_hours\_worked', '')()])

def \_update\_csv(self, employee):

with open('employees.csv', 'w', newline='') as file:

writer = csv.writer(file)

for employee in self.employees:

writer.writerow([employee.get\_name(), employee.get\_age(), employee.get\_salary(),

getattr(employee, 'get\_department', '')(), getattr(employee, 'get\_hours\_worked', '')()])

def \_save\_all\_to\_csv(self):

with open('employees.csv', 'w', newline='') as file:

writer = csv.writer(file)

for employee in self.employees:

writer.writerow([employee.get\_name(), employee.get\_age(), employee.get\_salary(),

getattr(employee, 'get\_department', '')(), getattr(employee, 'get\_hours\_worked', '')()])

system = EmployeeManagementSystem()

while True:

print("\nEmployee Management System")

print("1. Add Employee")

print("2. Display Employees")

print("3. Update Employee Information")

print("4. Delete Employee")

print("5. Exit")

choice = input("Enter your choice: ")

if choice == "1":

name = input("Enter employee name: ")

age = int(input("Enter employee age: "))

salary = float(input("Enter employee salary: "))

employee\_type = input("Enter employee type (Manager/Worker): ")

if employee\_type.lower() == "manager":

department = input("Enter department: ")

employee = Manager(name, age, salary, department)

elif employee\_type.lower() == "worker":

hours\_worked = int(input("Enter hours worked: "))

employee = Worker(name, age, salary, hours\_worked)

system.add\_employee(employee)

print("Employee added successfully.")

elif choice == "2":

print("\nEmployees:")

system.display\_employees()

elif choice == "3":

name = input("Enter employee name: ")

attribute = input("Enter attribute to update (name/age/salary/department/hours\_worked): ")

value = input("Enter new value: ")

system.update\_employee(name, attribute, value)

print("Employee information updated successfully.")

elif choice == "4":

name = input("Enter employee name to delete: ")

system.delete\_employee(name)

print("Employee deleted successfully.")

elif choice == "5":

print("Exiting program...")

break

else:

print("Invalid choice. Please enter a number between 1 and 5.")