1 Classes to Implement:

Person Class:

- Attributes:
    - name
    - money
    - mood
    - healthRate
- Methods:
    - sleep(hours)
    - eat(meals)
    - buy(items)

Employee Class (inherits from Person):

- Attributes:
    - id
    - car (Car object)
    - email
    - salary
    - distanceToWork
- Methods:
    - work(hours)
    - drive(distance)
    - refuel(gasAmount)
    - send_mail(to, subject, body)

Car Class:

- Attributes:
    - name
    - fuelRate (0 → 100)
    - velocity (0 → 200)
- Methods:
    - run(velocity, distance)
    - stop(remain_distance)

Office Class:

- Attributes:
    - name
    - employees (list of Employee objects)
- Methods:
    - get_all_employees()
    - get_employee(empId)
    - hire(Employee)
    - fire(empId)
    - deduct(empId, deduction)
    - reward(empId, reward)
    - check_lateness(empId, moveHour)
    - calculate_lateness(targetHour, moveHour, distance, velocity) [Static Method]
    - employeesNum (class variable)
    - change_emps_num(num) [Class Method]

2 Business Rules & Logic:

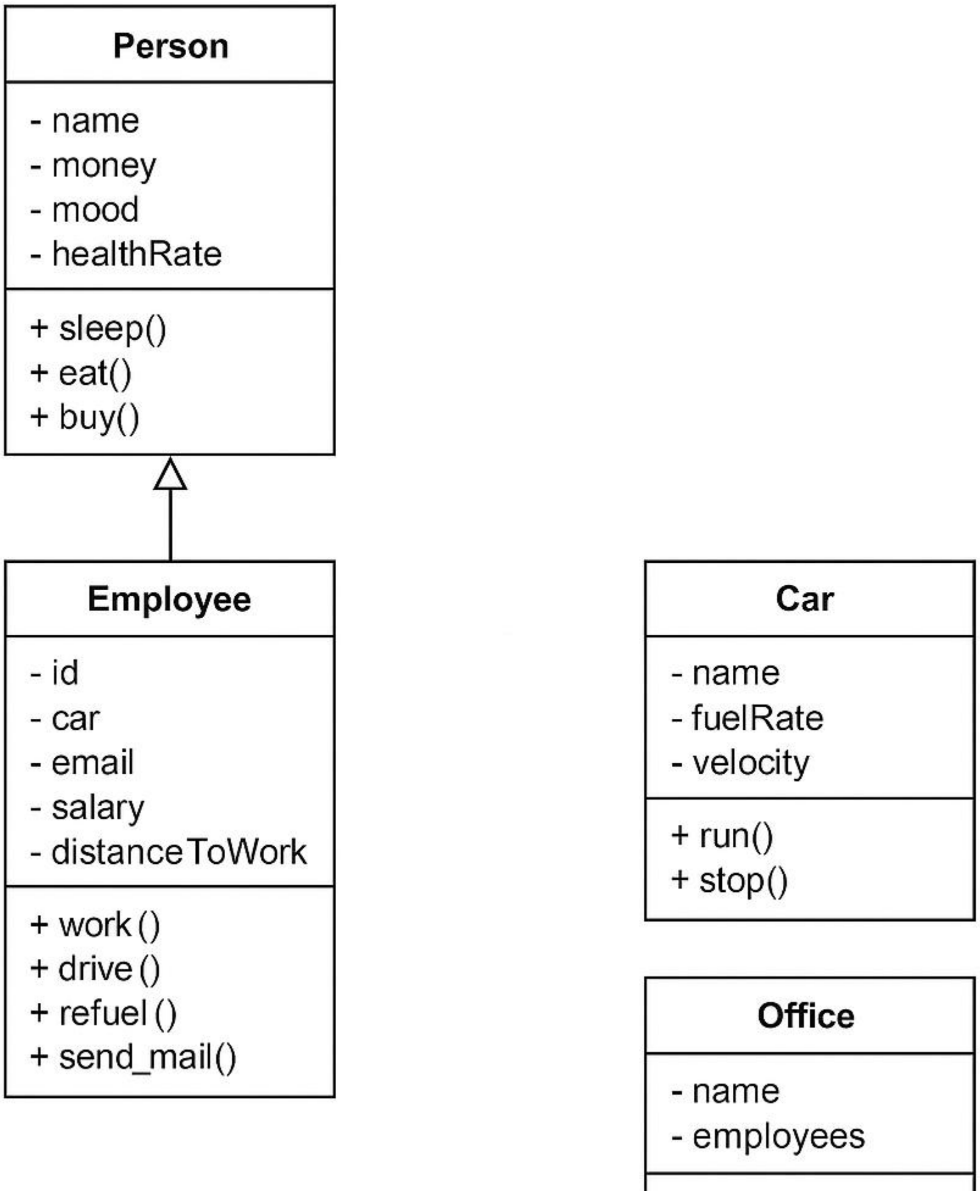
- **Fuel consumption decreases by 10% per 10 km.**
- **Employee is late if he arrives after 9:00 AM.**
- Salary is deducted by 10 L.E if late, rewarded by +10 L.E if on time.
- Car’s velocity must always be between 0 and 200 km/h.
- Car’s fuelRate must always be between 0 and 100%.

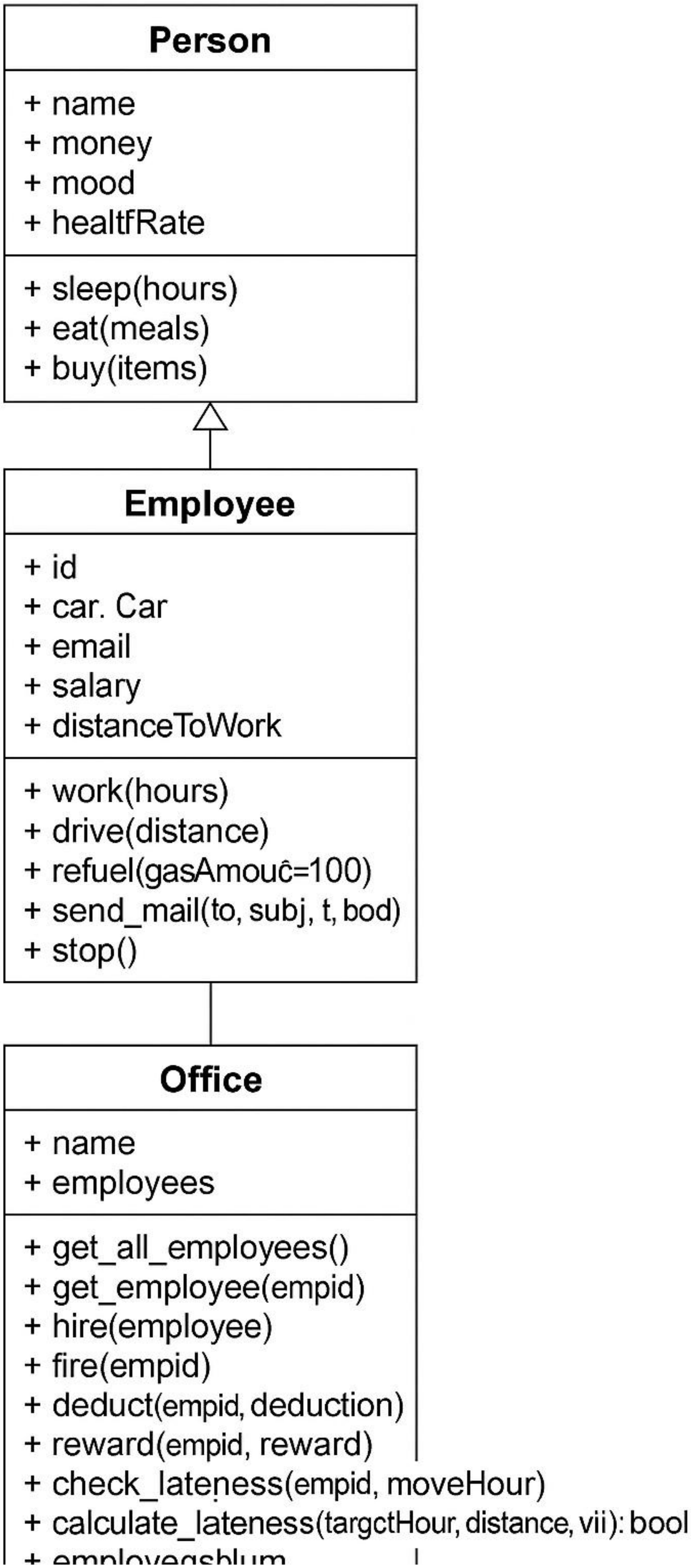
3 Simulation Scenario Example:

- Samy is an employee in ITI.
- He drives his Fiat 128 car every day to the ITI Smart Village Office.
- Distance from home to office is 20 km.
- The car consumes fuel per distance.
- The office checks if Samy is late or on time.
- Samy’s salary is adjusted based on his punctuality.

4 Optional Enhancements:

- Add multiple employees.
- Add weekend logic (Samy doesn’t work on weekends).
- Handle refueling automatically.
- Add Unit Tests for methods.


Person
- name
- money
- mood
- healthRate
+ sleep()
+ eat()
+ buy()
Employee
- id
- car
- email
- salary
- distanceToWork
+ work ()
+ drive ()
+ refuel ()
+ send_mail()
Car
- name
- fuelRate
- velocity
+ run()
+ stop()
Office
- name
- employees


Person
+ name
+ money
+ mood
+ healtfRate
+ sleep(hours)
+ eat(meals)
+ buy(items)
Employee
+ id
+ car. Car
+ email
+ salary
+ distanceToWork
+ work(hours)
+ drive(distance)
+ refuel(gasAmouĉ=100)
+ send_mail(to, subj, t, bod)
+ stop()
Office
+ name
+ employees
+ get_all_employees()
+ get_employee(empid)
+ hire(employee)
+ fire(empid)
+ deduct(empid, deduction)
+ reward(empid, reward)
+ check_lateness(empid, moveHour)
+ calculate_lateness(targctHour, distance, vii): bool
+ employeqshlum