PSEUDOCODE

## TO INITIALISE PUMPS

* CREATE object to hold new instance of Pumps list
* SET master Pumps list TO newly created object
* CREATE new pump object reference for the current instance of the Pump class
* LOOP 9 times, once per item
  + CREATE new Pump object
  + ADD new instance to master list of pumps
* END LOOP

## TO INITIALISE VEHICLES

* CREATE object to hold new instance of Vehicles list
* SET vehicles master list TO newly created object
* CREATE new timer object reference
* CREATE new timer object
* SET timer object reference TO newly created timer object
* CALL method to generate a random integer in the given
* SET timer interval TO random number returned from method
* CREATE new vehicle when timer interval has elapsed
* ENABLE timer to auto-reset
* ENABLE timer
* START timer

## TO INITIALISE THE TRANSACTION LOG FILE

* IF file with given path exists
  + DELETE file
* END IF

## TO INITIALISE THE PROGRAM ON STARTUP

* CALL method to initialise file
* CALL method to initialise pumps
* CALL method to initialise vehicles

## TO DRAW VEHICLE QUEUE ON CONSOLE

* CREATE new object reference for current Vehicle
* LOOP over every item in Vehicles master list
  + SET current vehicle BY accessing position ‘i’ from Vehicles list
  + PRINT carID to console
  + PRINT vehicle type to console
  + PRINT fuel type to console
* END LOOP

## TO DRAW FORECOURT ON CONSOLE

* CREATE new object reference for current Pump
* LOOP over every pump, once per item
  + CHECK if current pump is available
    - IF yes
    - IF no
  + END if
* END loop

## TO DRAW UI ON CONSOLE

* CALL method to display vehicle queue on console
* CALL method to display forecourt on console
* CALL method to display counters on console

## TO DISPLAY USER MENU ON STARTUP

* DECLARE repeat AS Boolean = true
* DO (repeat)
  + TRY
    - DECLARE user\_choice AS integer
    - SET user\_choice TO user input
    - CHECK if user\_choice = 2
      * IF yes
        + CALL method to use default values
        + SET repeat = false
      * IF no
        + CALL method to display error message
    - END IF
  + CATCH
    - CALL method to display error message on console
  + END TRY
* END loop

## TO SET DEFAULT PROGRAM VALUES

* SET litresDispensedPerSecond TO 1.5
* SET unleadedPrice TO 1.249
* SET dieselPrice TO 1.349
* SET lpgPrice TO 0.669

## TO DISPLAY STARTUP HOME PAGE TO USER

* CALL method to draw app title
* CALL method to display welcome message on console
* CALL method to display menu on console

## TO RUN ON PROGRAM STARTUP

* SET console window size TO 125x30
* CALL method to display startup home screen to user
* CALL method to initialise program
* CREATE new timer object reference
* CREATE new timer object
* SET timer object reference TO newly created timer object
* SET timer interval TO 1500 milliseconds
* CALL method to run program loop when timer has elapsed
* ENABLE timer to auto-reset
* ENABLE timer
* START timer

## TO RUN PROGRAM LOOP

* CLEAR console
* CALL method to draw UI onto console
* CALL method to assign vehicle to a pump if available

## TO GENERATE A RANDOM FUEL TYPE FOR A CAR

* DECLARE new list of strings with three items (Unleaded, Diesel, LPG)
* CREATE new random object reference
* CREATE new instance of Random class
* SET random object reference TO newly created object
* DECLARE index as integer
* CALL next random number method to generate a number less than the number of items in the list
* SET index TO number returned by random number method
* RETURN value at index in list

## TO GENERATE A RANDOM FUEL TYPE FOR A VAN

* DECLARE new list of strings with two items (Diesel, LPG)
* CREATE new random object reference
* CREATE new instance of Random class
* SET random object reference TO newly created object
* DECLARE index as integer
* CALL next random number method to generate a number less than the number of items in the list
* SET index TO number returned by random number method
* RETURN value at index in list

## TO GENERATE A RANDOM VEHICLE TYPE

* DECLARE new list of strings with three items (Car, Van, HGV)
* CREATE new random object reference
* CREATE new instance of Random class
* SET random object reference TO newly created object
* DECLARE index as integer
* CALL next random number method to generate a number less than the number of items in the list
* SET index TO number returned by random number method
* RETURN value at index in list

## TO GENERATE A RANDOM TIME AT WHICH A VEHICLE SHOULD BE CREATED

* CREATE new random object reference
* CREATE new instance of Random class
* SET random object reference TO newly created object
* CALL next random number function to generate a number between 1500 and 2200 inclusive
* RETURN randomly generated number

## TO GENERATE A RANDOM TIME IN WHICH A VEHICLE MUST BE SERVICED UPON CREATION

* CREATE new random object reference
* CREATE new instance of Random class
* SET random object reference TO newly created object
* CALL next random number function to generate a number between 1000 and 2000 inclusive
* RETURN randomly generated number

## TO CREATE A VEHICLE

* DECLARE random vehicle type as string
* CALL method to generate random vehicle type
* SET random vehicle type TO string returned by random vehicle type generator method
* CREATE new vehicle object
* ADD vehicle object to master Vehicle list

## TO CHECK IF A GIVEN PUMP IS AVAILABLE

* CHECK if a given pump is available
  + IF yes
    - RETURN true if null
  + IF no
    - RETURN false if not null
* END IF

## TO REMOVE A VEHICLE FROM THE QUEUE IF NOT SERVICED

* REMOVE item 0 from the Vehicles master list
* INCREMENT counter #5

## TO ASSIGN A VEHICLE TO A PUMP

* CREATE new object reference for current Vehicle
* CREATE new object references for firstLanePump, middleLanePump & endLanePump
* IF no vehicles in master list
  + RETURN
* END IF
* LOOP three times, once per row (row = 0, 1, 2 inclusive)
  + SET firstLanePump BY accessing position ‘i’ from Pumps list
  + CHECK if firstLanePump is available
    - IF yes
      * SET middleLanePump BY accessing position ‘i’ from Pumps list
      * CHECK if middleLanePump is available
        + IF yes

SET endLanePump BY accessing position ‘i’ from Pumps list

CHECK if endLanePump is available

IF yes

SET current vehicle TO first item in vehicles master list

REMOVE item from vehicles master list

ASSIGN current vehicle to endLanePump

STOP removalTimer

SET pump number TO endLanePump

IF no

SET current vehicle TO first item in vehicles master list

REMOVE item from vehicles master list

ASSIGN current vehicle to middleLanePump

STOP removalTimer

SET pump number TO middleLanePump

END IF

* + - * + IF no

SET current vehicle TO first item in vehicles master list

REMOVE item from vehicles master list

ASSIGN current vehicle to firstLanePump

STOP removalTimer

SET pump number TO firstLanePump

* + - * END IF
  + END IF
  + IF no vehicles in master list
    - BREAK loop
  + END IF
* END row loop

## TO HANDLE EVENTS WHEN THE VEHICLE HAS BEEN ASSIGNED TO A PUMP

* CREATE new object reference for current Vehicle
* CALL method to update the total number of litres of fuel dispensed
* CREATE new timer object reference
* CREATE new timer object
* SET timer object reference TO newly created timer object
* SET timer interval TO current vehicle’s refuel time
* CALL method to release vehicle when timer has elapsed
* SET timer auto-reset TO false
* ENABLE timer
* START timer

## TO UPDATE TOTAL LITRES OF FUEL DISPENSED

* CREATE new object reference for current Vehicle
* CHECK current vehicle’s fuel type
  + IF unleaded
    - UPDATE litresDispensedUnleaded
  + IF diesel
    - UPDATE litresDispensedDiesel
  + IF LPG
    - UPDATE litresDispensedLPG
* END IF

## ONCE A VEHICLE HAS BEEN SERVICED

* CALL method to log transaction details
* SET current vehicle TO null
* CALL method to show transaction details
* INCREMENT vehicles serviced

## TO LOG TRANSACTION DETAILS

* CREATE new object reference for current Vehicle
* CREATE new object reference for current pump number
* TRY
  + CREATE new stream writer object given a path
  + SET append TO true
  + CHECK current vehicle’s fuel type
    - IF unleaded
      * WRITE to file
    - IF diesel
      * WRITE to file
    - IF LPG
      * WRITE to file
  + END IF
  + CLOSE stream writer object reference
* CATCH
  + PRINT error to console
* END TRY

## TO SHOW TRANSACTION DETAILS

* CREATE new object reference for current Vehicle
* CREATE new object reference for current pump number
* TRY
  + CREATE new stream reader object given a path
  + WHILE end of file not reached
    - READ line
    - ADD each line to list
    - READ next line
  + END WHILE
  + PRINT last transaction on console
  + CLOSE stream reader object reference
* CATCH
  + PRINT error to console
* END TRY

## VEHICLE CLASS

* INCREMENT nextCarID
* SET carID TO nextCarID
* SET vehicleType TO string parameter
* CREATE new removalTimer object reference
* CREATE new timer object
* SET removalTimer object reference TO newly created timer object
* CALL method to randomly generate queuing time of vehicle
* SET timer interval TO random number returned by method
* CALL method to remove vehicle from queue when timer elapsed
* SET timer auto-reset TO false
* ENABLE timer
* START timer
* CHECK type of vehicle
  + IF car
    - CALL method to generate random type of fuel for a car
    - SET fuelType TO value returned by method
    - SET maxTankCapacity TO 40 litres
    - CALCULATE quarter tank capacity
    - SET quarterTank to quarterTankCapacity
    - CAST TO integer
    - CREATE new random object reference
    - CREATE new instance of Random class
    - SET random object reference TO newly created object
    - DECLARE currentTankCapacity as double
    - CALL next random number method to generate a number less than the quarter tank capacity
    - SET currentTankCapacity TO number returned by random number method
    - CALCULATE number of litres required to fill tank
      * maxTankCapacity – currentTankCapacity
    - CALCULATE refuel time
      * Litres until full / litres dispensed per second \* 1000
  + IF van
    - CALL method to generate random type of fuel for a van
    - SET fuelType TO value returned by method
    - SET maxTankCapacity TO 80 litres
    - CALCULATE quarter tank capacity
    - SET quarterTank to quarterTankCapacity
    - CAST TO integer
    - CREATE new random object reference
    - CREATE new instance of Random class
    - SET random object reference TO newly created object
    - DECLARE currentTankCapacity as double
    - CALL next random number method to generate a number less than the quarter tank capacity
    - SET currentTankCapacity TO number returned by random number method
    - CALCULATE number of litres required to fill tank
      * maxTankCapacity – currentTankCapacity
    - CALCULATE refuel time
      * Litres until full / litres dispensed per second \* 1000
  + IF HGV
    - SET fuelType TO diesel
    - SET fuelType TO value returned by method
    - SET maxTankCapacity TO 150 litres
    - CALCULATE quarter tank capacity
    - SET quarterTank to quarterTankCapacity
    - CAST TO integer
    - CREATE new random object reference
    - CREATE new instance of Random class
    - SET random object reference TO newly created object
    - DECLARE currentTankCapacity as double
    - CALL next random number method to generate a number less than the quarter tank capacity
    - SET currentTankCapacity TO number returned by random number method
    - CALCULATE number of litres required to fill tank
      * maxTankCapacity – currentTankCapacity
    - CALCULATE refuel time
      * Litres until full / litres dispensed per second \* 1000
* END IF