**Problem #2 - Fuel Station Design**:

class FuelStation:  
 def \_\_init\_\_(self, diesel: int, petrol: int, electric: int):  
 # Initialize the available slots and total slots for each fuel type  
 self.total\_slots = {  
 "diesel": diesel,  
 "petrol": petrol,  
 "electric": electric  
 }  
 self.available\_slots = {  
 "diesel": diesel,  
 "petrol": petrol,  
 "electric": electric  
 }  
  
 def fuel\_vehicle(self, fuel\_type: str) -> bool:  
 # Check if there is an available slot for the specified fuel type  
 if self.available\_slots.get(fuel\_type, 0) > 0:  
 self.available\_slots[fuel\_type] -= 1 # Occupy a slot  
 return True  
 return False  
  
 def open\_fuel\_slot(self, fuel\_type: str) -> bool:  
 # Check if any slot can be opened for the specified fuel type  
 if self.available\_slots.get(fuel\_type, 0) < self.total\_slots.get(fuel\_type, 0):  
 self.available\_slots[fuel\_type] += 1 # Release a slot  
 return True  
 return False  
fuel\_station = FuelStation(diesel=2, petrol=2, electric=1)  
output = []  
output.append(fuel\_station.fuel\_vehicle("diesel"))  
output.append(fuel\_station.fuel\_vehicle("petrol"))  
output.append(fuel\_station.fuel\_vehicle("diesel"))  
output.append(fuel\_station.fuel\_vehicle("electric"))  
output.append(fuel\_station.fuel\_vehicle("diesel"))  
output.append(fuel\_station.open\_fuel\_slot("diesel"))  
output.append(fuel\_station.fuel\_vehicle("diesel"))  
output.append(fuel\_station.open\_fuel\_slot("electric"))  
output.append(fuel\_station.open\_fuel\_slot("electric"))  
print("[" + " ".join(map(lambda x: str(x).lower(), output)) + "]")

output:

**[true, true, true, true, false, true, true, true, false]**