

## Exercise 2

1. Define the required classes, methods and properties based on CarSalesDataforReports Excel file.

```
class Clients:
    #Attributes
    clientID=0
    clientName=""
    address1=""
    address2=""
    town=""
    country=""
    postCode=""
    region=""
    outerPostode=""
    countryID=0
    clientType=""
    clientSize=""
    clientSince=""
    isCreditWorthy = False
    isDealer = False

    #Methods

    def __init__(self,clientID, clientName, address1, address2, town, country, postCode,
                region, outerPostode, countryID, clientType, clientSize, clientSince,
                isCreditWorthy, isDealer):
        self.clientID = clientID
        self.clientName = clientName
        self.address1 = address1
        self.address2 = address2
        self.town = town
        self.country = country
        self.postCode = postCode
        self.region = region
        self.outerPostode = outerPostode
        self.countryID = countryID
        self.clientType = clientType
        self.clientSize = clientSize
        self.clientSince = clientSince
        self.isCreditWorthy = isCreditWorthy
        self.isDealer = isDealer

    #Gets
    def getClientID(self):
        return self.clientID

    def getClientName(self):
        return self.clientName

    def getAddress1(self):
        return self.address1

    def getAddress2(self):
        return self.address2
```

```
def getTown(self):
    return self.town

def getCountry(self):
    return self.country

def getPostCode(self):
    return self.postCode

def getRegion(self):
    return self.region

def getOuterPostode(self):
    return self.outerPostode

def getCountryID(self):
    return self.countryID

def getClientType(self):
    return self.clientType

def getClientSize(self):
    return self.clientSize

def getClientSince(self):
    return self.clientSince

def getIsCreditWorthy(self):
    return self.isCreditWorthy

def getIsDealer(self):
    return self.isDealer

#Sets
def setClientID(self, newClientID):
    self.clientID = newClientID

def setClientName(self, newClientName):
    self.clientName = newClientName

def setAddress1(self, newAddress1):
    self.address1 = newAddress1

def setAddress2(self, newAddress2):
    self.address2 = newAddress2

def setTown(self, newTown):
    self.town = newTown

def setCountry(self, newCountry):
    self.country = newCountry

def setPostCode(self, newPostCode):
    self.postCode = newPostCode

def setRegion(self, newRegion):
    self.region = newRegion

def setOuterPostode(self, newOuterPostode):
```

```

        self.outerPostode = newOuterPostode

def setCountryID(self, newCountryID):
    self.countryID = newCountryID

def setClientType(self, newClientType):
    self.clientType = setClientType

def setClientSize(self, newClientSize):
    self.clientSize = newClientSize

def setClientSince(self, newClientSince):
    self.clientSince = newClientSince

def setIsCreditWorthy(self, newIsCreditWorthy):
    self.isCreditWorthy = newIsCreditWorthy

def setIsDealer(self, newIsDealer):
    self.isDealer = newIsDealer

```

```

class Colors:
    #Attributes
    colorID=0
    color=""

    #Methods

    def __init__(self, colorID, color):
        self.colorID = colorID
        self.color = color

#Gets
def getColorID(self):
    return self.colorID

def getColor(self):
    return self.color

#Sets
def setColorID(self, newColoID):
    self.colorID = newColoID

def setColor(self, newColor):
    self.color = newColor

```

```

class Countries:
    #Attributes
    countryID=0
    countryName=""
    countryISOCODE=""

    #Methods

    def __init__(self, countryID, countryName, countryISOCODE):
        self.countryID = countryID

```

```

    self.countryName = countryName
    self.countryISOCODE = countryISOCODE

#Gets
def getCountryID(self):
    return self.countryID

def getCountryName(self):
    return self.countryName

def getCountryISOCODE(self):
    return self.countryISOCODE

#Sets
def setCountryID(self, newCountryID):
    self.countryID = newCountryID

def setCountryName(self, newCountryName):
    self.countryName = newCountryName

def setCountryISOCODE(self, newCountryISOCODE):
    self.countryISOCODE = newCountryISOCODE

class Stock:
    #Attributes
    stockID=0
    make=""
    model=""
    colorID=""
    vehicleType=""
    costPrice=0.0
    spareParts=0.0
    laborCost=0.0
    registrationDate=""
    mileage=0.0
    purchaseDate=""
    vehicleAgeInYears=0

    #Methods

def __init__(self, stockID, make, model, colorID, vehicleType, costPrice, spareParts,
             laborCost, registrationDate, mileage, purchaseDate, vehicleAgeInYears):
    self.stockID=stockID
    self.make=make
    self.model=model
    self.colorID=colorID
    self.vehicleType=vehicleType
    self.costPrice=costPrice
    self.spareParts=spareParts
    self.laborCost=laborCost
    self.registrationDate=registrationDate
    self.mileage=mileage
    self.purchaseDate=purchaseDate
    self.vehicleAgeInYears=vehicleAgeInYears

def infoBasic(self):
    print(f"Make: {self.make}, Model: {self.model}, vehicle type: {self.vehicleType}")

```

```
#Gets
def getStockID(self):
    return self.stockID

def getMake(self):
    return self.make

def getModel(self):
    return self.model

def getColorID(self):
    return self.colorID

def getVehicleType(self):
    return self.vehicleType

def getCostPrice(self):
    return self.costPrice

def getSpareParts(self):
    return self.spareParts

def getLaborCost(self):
    return self.laborCost

def getRegistrationDate(self):
    return self.registrationDate

def getMileage(self):
    return self.mileage

def getPurchaseDate(self):
    return self.purchaseDate

def getVehicleAgeInYears(self):
    return self.vehicleAgeInYears

#Sets
def SetStockID(self, newStockID):
    self.stockID = newStockID

def SetMake(self, newMake):
    self.make = newMake

def SetModel(self, newModel):
    self.model = newModel

def SetColorID(self, newColorID):
    Self.colorID = newColorID

def SetVehicleType(self, newVehicleType):
    self.vehicleType = newVehicleType

def SetCostPrice(self, newCostPrice):
    self.costPrice = newCostPrice

def SetSpareParts(self, newSpareParts):
    self.spareParts = newSpareParts
```

```
def SetLaborCost(self, newLaborCost):
    self.laborCost = newLaborCost

def SetRegistrationDate(self, newRegistrationDate):
    self.registrationDate = newRegistrationDate

def SetMileage(self, newMileage):
    self.mileage = newMileage

def SetPurchaseDate(self, newPurchaseDate):
    self.purchaseDate = newPurchaseDate

def SetVehicleAgeInYears(self, newVehicleAgeInYears):
    self.vehicleAgeInYears = newVehicleAgeInYears
```

```
class Invoices:
    #Attributes
    invoiceID=0
    invoiceNumber=""
    clientID=0
    invoiceDate=""
    totalDiscount=0.0
    deliveryCharge=0.0
    invoicedDateKey=0

    #Methods

    def __init__(self, invoiceID, invoiceNumber, clientID, invoiceDate, totalDiscount,
        deliveryCharge, invoiceDateKey):
        self.invoiceID=invoiceID
        self.invoiceNumber=invoiceNumber
        self.clientID=clientID
        self.invoiceDate=invoiceDate
        self.totalDiscount=totalDiscount
        self.deliveryCharge=deliveryCharge
        self.invoiceDateKey=invoiceDateKey

    #Gets
    def getInvoiceID(self):
        return self.invoiceID

    def getInvoiceNumber(self):
        return self.invoiceNumber

    def getClientID(self):
        return self.clientID

    def getInvoiceDate(self):
        return self.invoiceDate

    def getTotalDiscount(self):
        return self.totalDiscount
```

```
def getDeliveryCharge(self):
    return self.deliveryCharge

def getInvoiceDateKey(self):
    return self.invoiceDateKey

#Sets
def setInvoiceID(self, newInvoiceID):
    self.invoiceID = newInvoiceID

def setInvoiceNumber(self, newInvoiceNumber):
    self.invoiceNumber = newInvoiceNumber

def setClientID(self, newClientID):
    self.clientID = newClientID

def setInvoiceDate(self, newInvoiceDate):
    self.invoiceDate = newInvoiceDate

def setTotalDiscount(self, newTotalDiscount):
    self.totalDiscount = newTotalDiscount

def setDeliveryCharge(self, newDeliveryCharge):
    self.deliveryCharge = newDeliveryCharge

def setInvoiceDateKey(self, newInvoiceDateKey):
    self.invoiceDateKey = newInvoiceDateKey
```

```
class Invoicelines:
    #Attributes
    invoiceLineID=0
    invoiceID=0
    stockID=0
    salePrice=0.0
    lineItem=0

    #Methods

    def __init__(self, invoiceLineID, invoiceID, stockID, salePrice, lineItem):
        self.invoiceLineID=invoiceLineID
        self.invoiceID=invoiceID
        self.stockID=stockID
        self.salePrice=salePrice
        self.lineItem=lineItem

    #Gets
    def getInvoiceLineID(self):
        return self.invoiceLineID

    def getInvoiceID(self):
        return self.invoiceID

    def getStockID(self):
```

```

        return self.stockID

def getSalePrice(self):
    return self.salePrice

def getLineItem(self):
    return self.lineItem

#Gets
def SetInvoiceLineID(self, newInvoiceLineID):
    self.invoiceLineID = newInvoiceLineID

def SetInvoiceID(self, newInvoiceID):
    self.invoiceID = newInvoiceID

def SetStockID(self, newStockID):
    self.stockID = newStockID

def SetSalePrice(self, newSalePrice):
    self.salePrice = newSalePrice

def SetLineItem(self, newLineItem):
    self.lineItem = newLineItem


class DateDimension:
    #Attributes
    dateKey=0
    year=0
    monthNum=0
    monthFull=""
    monthAbbr=""
    quarterNum=0
    quarterFull=""
    quarterAbbr=""
    yearAndQuarterNum=0
    quarterAbbrAndYear=""
    monthAbbrAndYear=""
    monthAndYear=""
    monthName=""
    monthNameAbbr=""
    quarterAndYear=""
    quarterAndYearAbbr2=""
    yearAndMonthNum=0

    #Methods

    def __init__(self, dateKey, year, monthNum, monthFull, monthAbbr, quarterNum,
                quarterFull, quarterAbbr, yearAndQuarterNum, quarterAbbrAndYear,
                monthAbbrAndYear, monthAndYear, monthName, monthNameAbbr, quarterAndYear,
                quarterAndYearAbbr2, yearAndMonthNum):
        Self.dateKey=dateKey
        Self.year=year
        Self.monthNum=monthNum
        Self.monthFull=monthFull
        Self.monthAbbr=monthAbbr
        Self.quarterNum=quarterNum

```



```
Self.quarterFull=quarterFull
Self.quarterAbbr=quarterAbbr
Self.yearAndQuarterNum=yearAndQuarterNum
Self.quarterAbbrAndYear=quarterAbbrAndYear
Self.monthAbbrAndYear=monthAbbrAndYear
Self.monthAndYear=monthAndYear
Self.monthName=monthName
Self.monthNameAbbr=monthNameAbbr
Self.quarterAndYear=quarterAndYear
Self.quarterAndYearAbbr2=quarterAndYearAbbr2
Self.yearAndMonthNum=yearAndMonthNum
```

#Gets

```
def getDateKey(self):
    return self.dateKey
```

```
def getYear(self):
    return self.year
```

```
def getMonthNum(self):
    return self.monthNum
```

```
def getMonthFull(self):
    return self.monthFull
```

```
def getMonthAbbr(self):
    return self.monthAbbr
```

```
def getQuarterNum(self):
    return self.quarterNum
```

```
def getQuarterFull(self):
    return self.quarterFull
```

```
def getQuarterAbbr(self):
    return self.quarterAbbr
```

```
def getYearAndQuarterNum(self):
    return self.yearAndQuarterNum
```

```
def getQuarterAbbrAndYear(self):
    return self.quarterAbbrAndYear
```

```
def getMonthAbbrAndYear(self):
    return self.monthAbbrAndYear
```

```
def getMonthAndYear(self):
    return self.monthAndYear
```

```
def getMonthName(self):
    return self.monthName
```

```
def getMonthNameAbbr(self):
    return self.monthNameAbbr
```

```
def getQuarterAndYear(self):
    return self.quarterAndYear
```

```
def getQuarterAndYearAbbr2(self):
    return self.quarterAndYearAbbr2

def getYearAndMonthNum(self):
    return self.yearAndMonthNum

#Sets
def setDateKey(self, newDateKey):
    self.dateKey = newDateKey

def setYear(self, newYear):
    self.year = newYear

def setMonthNum(self, newMonthNum):
    self.monthNum = newMonthNum

def setMonthFull(self, newMonthFull):
    self.monthFull = newMonthFull

def setMonthAbbr(self, newMonthAbbr):
    self.monthAbbr = newMonthAbbr

def setQuarterNum(self, newQuarterNum):
    self.quarterNum = newQuarterNum

def setQuarterFull(self, newQuarterFull):
    self.quarterFull = newQuarterFull

def setQuarterAbbr(self, newQuarterAbbr):
    self.quarterAbbr = newQuarterAbbr

def setYearAndQuarterNum(self, newYearAndQuarterNum):
    self.yearAndQuarterNum = newYearAndQuarterNum

def setQuarterAbbrAndYear(self, newQuarterAbbrAndYear):
    self.quarterAbbrAndYear = newQuarterAbbrAndYear

def setMonthAbbrAndYear(self, newMonthAbbrAndYear):
    self.monthAbbrAndYear = newMonthAbbrAndYear

def setMonthAndYear(self, newMonthAndYear):
    self.monthAndYear = newMonthAndYear

def setMonthName(self, newMonthName):
    self.monthName = newMonthName

def setMonthNameAbbr(self, newMonthNameAbbr):
    self.monthNameAbbr = newMonthNameAbbr

def setQuarterAndYear(self, newQuarterAndYear):
    self.quarterAndYear = newQuarterAndYear

def setQuarterAndYearAbbr2(self, newQuarterAndYearAbbr2):
    self.quarterAndYearAbbr2 = newQuarterAndYearAbbr2

def setYearAndMonthNum(self, newYearAndMonthNum):
    self.yearAndMonthNum = newYearAndMonthNum
```

2. Based on the point number one, define a class to include another kind of transportation vehicle. Include properties and methods that you consider required.

```
class Vehicle(Stock):  
    pass  
  
pickup = Vehicle(1, "Toyota", "Hilux", "Blanco", "Pickup", "1000", "2000",  
                 "3000", "20/1/2022", "4000", "20/1/2022", "2")  
  
pickup.infoBasic()  
  
Make: Toyota, Model: Hilux, vehicle type:  Pickup
```