

Part 1

StockItem

- StockCode : String

- Quantity : int

- Price : double

+StockItem(Code: String , quantity: Integer, price: Double)

+SetNewPrice(price: Double) : void

+getStockName() : String

+getStockDescription() : String

+addStock(StockLevel: Integer) : void

+sellStock(StockLevel: Integer) : Boolean

+getVAT() : Double

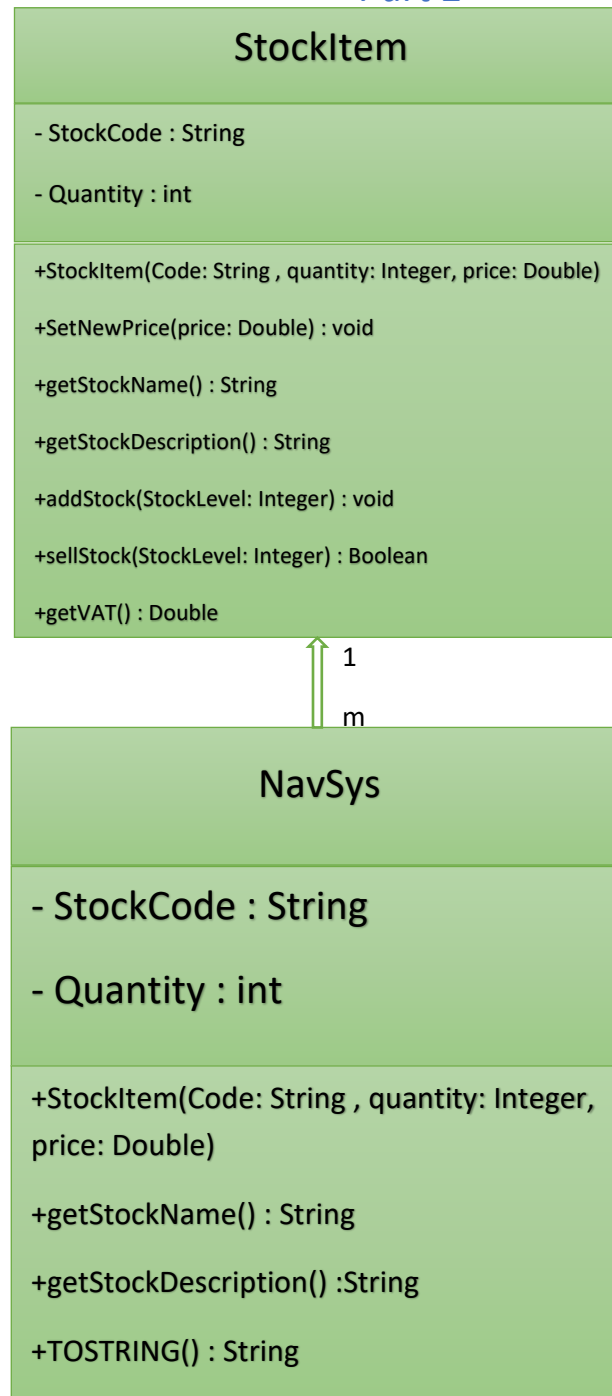
+PriceWithVAT(p : Double) : Double

+PriceWithoutVAT(p: Double) : Double

Test for Stock Item Class			
Test Case	Purpose	Expected Result	Actual results
1	Creating a stock with 10 units Unknown item, price 99.99 each, and item code W101	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 10	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 10
2	Increasing 10 more units in stock	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 20	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 20
3	Sold 2 units from stock	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 18	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 18
4	Set new price 100.99 per unit	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 100.99 PriceWithVAT: 118.66324999999999 Total unit in stock: 18	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 100.99 PriceWithVAT: 118.66324999999999 Total unit in stock: 18

5	Increasing 0 more units	Increasing 0 more units Increased item must be greater than or equal to one	The error was: Increased item must be greater than or equal to one
6	Testing getStockName	<i>Must output: Unknown stock name</i>	<i>Stock Type: Unknown Stock Name</i>
7	<i>Testing getStockDescription</i>	<i>Must output: Unknown stock Description</i>	<i>Description: Unknown Stock Description</i>
8	<i>getVAT</i>	<i>Return VAT value</i>	<i>Vat Value outputted- The VAT value was used in the program as predicted</i>
9	<i>PriceWithVAT</i>	<i>Return new price with added vat Price with VAT: 117.48825</i>	<i>Price with VAT: 117.48825</i>
10	<i>PriceWithoutVAT</i>	<i>Returns Raw Price Price without VAT: 100.99</i>	<i>Price without VAT: 100.99</i>

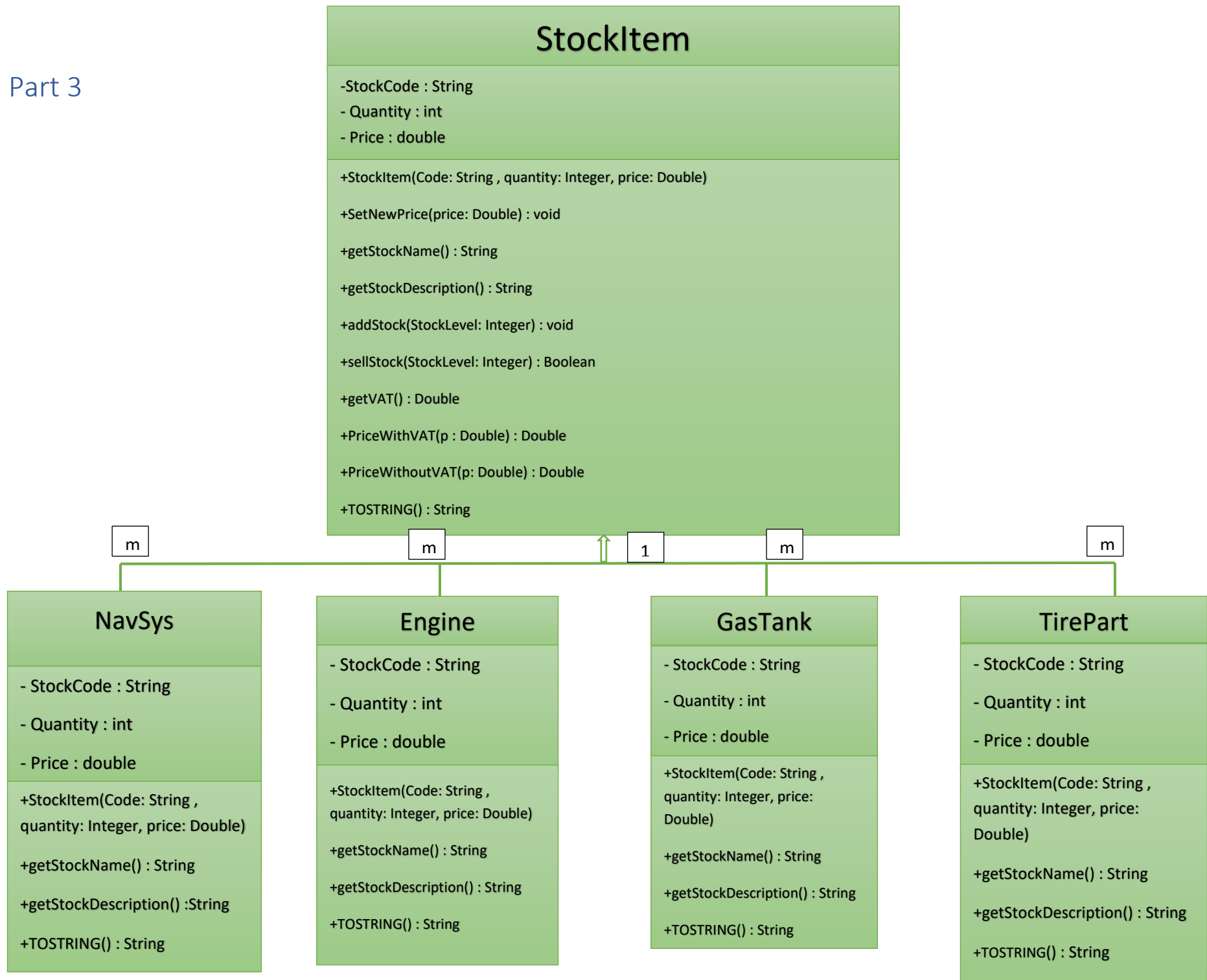
Part 2



Test for TestNavSys Class and NavSys class			
Test Case	Purpose	Expected Result	Actual Result
1	Creating a stock with 10 units Unknown item, price 99.99 each, and item code W101	Printing item stock information: Stock Type: Unknown Stock Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 10	Test successful Stock Type: Unknown Stock Name Description: Unknown Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 10
2	Increasing 10 more units in stock	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 20	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 20
3	Sold 2 units from stock	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 18	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 99.99 PriceWithVAT: 117.48825 Total unit in stock: 18
4	Set new price 100.99 per unit	Printing item stock information: Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 100.99 PriceWithVAT: 118.66324999999999 Total unit in stock: 18	Test successful Stock Type: Unknown Stock Name Description: Unknown Stock Description StockCode: W101 PriceWithoutVAT: 100.99 PriceWithVAT: 118.66324999999999 Total unit in stock: 18
5	Increasing 0 more units	Required output Increasing 0 more units	Actual Output Increasing 0 more units

		Increased item must be greater than or equal to one	Increased item must be greater than or equal to one
6	Override GetstockName() method	Method must be overridden to output; "Stock Type: Navigation system"	Correct output; Stock Type: Navigation system
7	Override getStockDescription() method	Method must be overridden to output; "Stock Type: GeoVision Sat Nav"	Correct output Stock Type: GeoVision Sat Nav
8	ToString method	Must print out information; Stock Type: Description: Stock Code Price without VAT: Price with VAT: Total unit in Stock:	Stock Type: Unknown Stock Name Description: Unknown Stock Description Stock Code: W101 Price without VAT: 99.99 Price with VAT: 117.48825 Total unit in Stock: 10

Part 3



Test for Classes : GasTank, Engine, TirePart and TestPolymorphism			
Test Case	Purpose	Expected result	Actual Result
1	To check if the array holds data correctly	Program outputs 3 sets of data for 3 of the subclasses GasTank, Engine and TirePart	No Errors, all 3 classes utilised, array functions correctly.
2	Test Data input for Engine (Scanner Input)	Data inputs correctly for stock code, stock price, stock amount.	Data is inputted correctly for all required inputs
3	Test Data input for GasTank (Scanner Input)	Data inputs correctly for stock code, stock price, stock amount.	Data is inputted correctly for all required inputs
4	Test Data input for TirePart(Scanner Input)	Data inputs correctly for stock code, stock price, stock amount.	Data is inputted correctly for all required inputs
5	Test variables for storing user inputted data, engineStockCodeInput engineStockAmountInput enginePriceInput	Variables should store data, this will be proven when the data outputs correctly when then program is run.	The data is output correctly when the program is run.
6	Test variables for storing user inputted data, gasTankStockCodeInput gasTankAmountInput gasTankPriceInput	Variables should store data, this will be proven when the data outputs correctly when then program is run.	The data is output correctly when the program is run.
7	Test variables for storing user inputted data, tirePartStockCodeInput tirePartStockAmountInput tirePartPriceInput	Variables should store data, this will be proven when the data outputs correctly when then program is run.	The data is output correctly when the program is run.
8	Scanner input object	Should create an object for the program to use allowing it to take user input from the keyboard	The program takes user input from keyboard
9	Test For loop for outputting data from all 3 classes	For loop should run 3 times allowing all classes to be displayed	Loop runs 3 times

10	Increasing 10 more units in stock GasTank class	Increasing 10 more units Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 12.0 Price with VAT: 14.1 Total unit in Stock: 23	Test successful Increasing 10 more units Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 12.0 Price with VAT: 14.1 Total unit in Stock: 23
11	Increasing 10 more units in stock for Engine Class	Increasing 10 more units Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 1234.0 Price with VAT: 1449.95 Total unit in Stock: 22	Test successful Increasing 10 more units Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 1234.0 Price with VAT: 1449.95 Total unit in Stock: 22
12	Increasing 10 more units in stock for TirePart Class	Increasing 10 more units Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 145.0 Price with VAT: 170.375 Total unit in Stock: 23	Test successful Increasing 10 more units Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 145.0 Price with VAT: 170.375 Total unit in Stock: 23
13	Sold 2 units for Engine Class	Sold 2 units Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 1234.0 Price with VAT: 1449.95 Total unit in Stock: 20	Test successful Sold 2 units Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 1234.0 Price with VAT: 1449.95 Total unit in Stock: 20

14	Sold 2 units for GasTank Class	Sold 2 units Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 12.0 Price with VAT: 14.1 Total unit in Stock: 21	Test successful Sold 2 units Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 12.0 Price with VAT: 14.1 Total unit in Stock: 21
15	Sold 2 units for Tire Class	Sold 2 units Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 145.0 Price with VAT: 170.375 Total unit in Stock: 21	Test successful Sold 2 units Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 145.0 Price with VAT: 170.375 Total unit in Stock: 21
16	Set new price 100 per unit for Engine class.	Set new price 100 per unit Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 20	Test successful Set new price 100 per unit Stock Type: Engine Description: v8 4 stroke Stock Code: nfrw Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 20
17	Set new price 100 per unit for GasTank class.	Set new price 100 per unit Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 21	Test successful Set new price 100 per unit Stock Type: Gas tank Description: 80 leters Stock Code: be11 Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 21

18	Set new price 100 per unit for TirePart class.	Set new price 100 per unit Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 21	Test successful Set new price 100 per unit Stock Type: Tire Description: Winter Tires Stock Code: fqw1 Price without VAT: 100.0 Price with VAT: 117.5 Total unit in Stock: 21
19	Increasing 0 more units for Engine Class	Increasing 0 more units Increased item must be greater than	Test Successful Increasing 0 more units Increased item must be greater than or equal to one
20	Increasing 0 more units for GasTank Class	Increasing 0 more units Increased item must be greater than	Test Successful Increasing 0 more units Increased item must be greater than or equal to one
21	Increasing 0 more units for TirePart Class	Increasing 0 more units Increased item must be greater than	Test Successful Increasing 0 more units Increased item must be greater than or equal to one
22	Override Stock code method Engine	Stock Code method is overridden	Data Outputted correctly
23	Override Stock code method GasTank	Stock Code method is overridden	Data Outputted correctly
24	Override Stock code method TirePart	Stock Code method is overridden	Data Outputted correctly
25	Override Stock Description Engine	Stock Description method is overridden	Data Outputted correctly
26	Override Stock Description GasTank	Stock Description method is overridden	Data Outputted correctly
27	Override Stock Description TirePart	Stock Description method is overridden	Data Outputted correctly
28	Override ToString Engine	ToString method overridden	Data Outputted correctly
29	Override ToString GasTank	ToString method overridden	Data Outputted correctly
30	Override ToString TirePart	ToString method overridden	Data Outputted correctly

31	Test constructor Engine	super(code, quantity, price) Variables must be in given to stock item correctly	Data handed over correctly program runs correctly
32	Test constructor GasTank	super(code, quantity, price); Variables must be in given to stock item correctly	Data Outputted correctly program runs correctly
33	Test constrictor TirePart	super(code, quantity, price); Variables must be in given to stock item correctly	Data handed over correctly program runs correctly