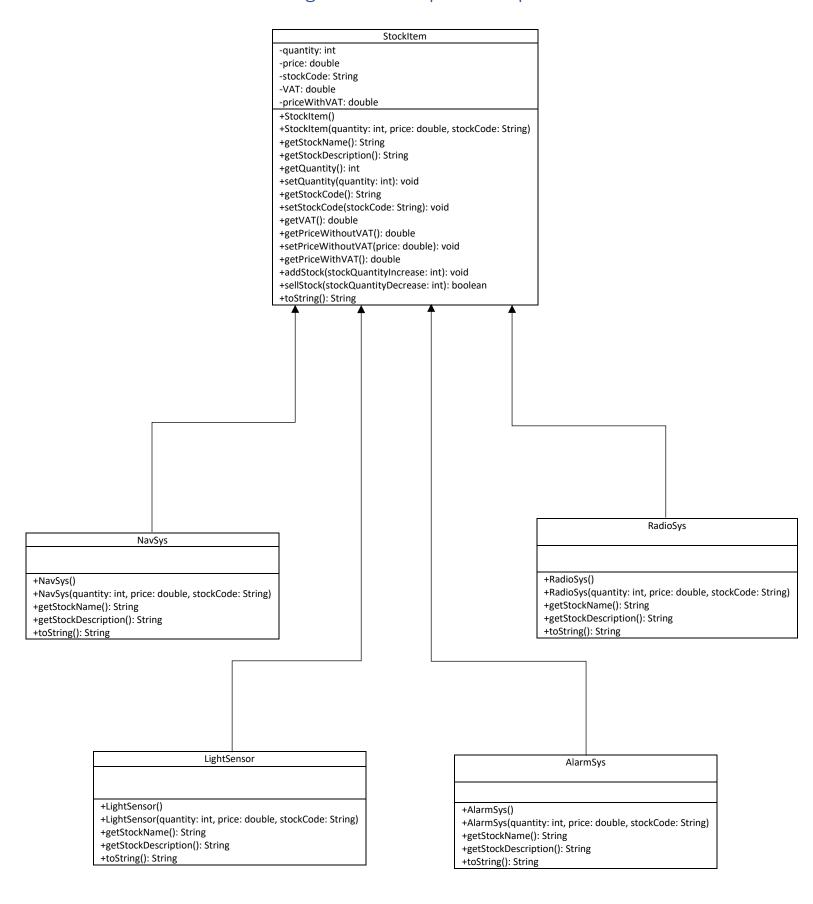
UML diagram for car parts shop



TestPolymorphism		
+itemInstance(item: StockItem): void		
+checkInitialQuantity(): int		
+checkInitialPrice(): double		
+checkCode(): String		

Test cases

Note: " $\it NavSys"$ class will be used as example in the test case scenarios below

Test case	Purpose	Expected result		
quantity = -1	Test an initial stock quantity which is	Error message		
	negative			
quantity = 0	Test an initial stock quantity which is 0	Error message		
quantity = 101	Test an initial stock quantity which	Error message		
(1.7)	exceeds the maximum stock quantity			
quantity = "k"	Test an initial stock quantity which is a	Error message		
	string			
quantity = 88.88	Test an initial stock quantity which is a	Error message		
	float/double	6		
quantity = 10	Test an initial stock quantity which is	No error message/Valid input		
	within the valid range			
	[1, 100]			
price = -1	Test an initial item price which is	Error message		
	negative			
price = 0	Test an initial item price which is 0	Error message		
price = "k"	Test an initial price which is a string	Error message		
price = 99.99	Test an initial item price which is	No error message/Valid input		
	positive			
stockCode = "NS101"	Test an initial stock code	No error message/Valid input		
After a valid initialisation				
	NavSys Class(current stock info)			
quantity = 10				
	price = 99.99			
stockCode = "NS101"				
	StockCode - NSIOI			
increaseQuantity = -1	Test increasing the stock by a negative	Error message/no stock		
·	value	increase		
		quantity = 10		
increaseQuantity = 0	Test increasing the stock by 0	Error message/no stock		
, ,	,	increase		
		quantity = 10		

increaseQuantity = 91	Test increasing the stock by a value	Error message/no stock
	which makes the stock exceed its	increase
	maximum quantity	quantity = 10
increaseQuantity = 101	Test increasing the stock by a value	Error message/no stock
	which makes the stock exceed its	increase
	maximum quantity	quantity = 10
IncreaseQuantity = "k"	Test increasing the stock by a string	Error message/no stock
	value	increase
		quantity = 10
IncreaseQuantity = 88.88	Test increasing the stock by a	Error message/no stock
	float/double value	increase
		quantity = 10
increaseQuantity = 20	Test increasing the stock by a valid	No error message/Valid stock
,	value	increase
		quantity = 30
	After a valid stock increase	, ,
	NavSys Class(current stock info) quantity = 30 price = 99.99 stockCode = "NS101"	
decreaseQuantity = -1	Test decreasing the stock by a	Error message/no stock
decrease Quartity = 1	negative value	reduction
	negative value	quantity = 30
decreaseQuantity = 0	Test decreasing the stock by 0	Error message/no stock
decrease Quartity = 0	rest decreasing the stock by o	reduction
		quantity = 30
decreaseQuantity = 31	Test decreasing the stock by a value	Error message/no stock
decrease Quartity = 31	greater than the current stock	reduction
	quantity	quantity = 30
decreaseQuantity = "k"	Test decreasing the stock by a string	Error message/no stock
decreaseQuartity = k	value	reduction
	value	quantity = 30
decreaseQuantity = 88.88	Test decreasing the stock by a	Error message/no stock
decreaseQualitity = 88.88	float/double value	reduction
	float/double value	
docroscoQuantity = 15	Tost decreasing the stock by a valid	quantity = 30 No error message/Valid stock
decreaseQuantity = 15	Test decreasing the stock by a valid value	reduction
	value	
		quantity = 15

After a valid stock reduction

NavSys Class(current stock info) quantity = 15 price = 99.99 stockCode = "NS101"

price = -100	Test setting a new price which is	Error message/no price update
	negative	price = 99.99
price = 0	Test setting a new price which is 0	Error message/no price update
		price = 99.99
price = "k"	Test setting a new price which is a	Error message/no price update
	string value	price = 99.99
price = 120	Test setting a new valid price	No error message/Valid price
		update
		price = 120

After setting new price

NavSys Class(current stock info) quantity = 15 price = 120 stockCode = "NS101"