**TESTING: TASK 1.2**

|  |  |  |
| --- | --- | --- |
| **TEST CASE** | **PURPOSE** | **EXPECTED RESULT** |
| **Initiation of Stock with Parameters:**  **(“String stockCode, int quantity, double price”)** | Initiating a new stock:  *“StockItem stock = new StockItem("W101", 10, 99.99)”*  Checks that the parameters given are printing correctly. | **Printing item stock information:**  Stock Type: Unknown Stock Name  Stock Description: Unknown Stock Description  Stock Code: W101  Price (excl. VAT): £99.99  Price (inc. VAT): £117.48825  Total Units In Stock: 10 |
| **Add > 100 Units of Stock** | Checks for valid input of adding stock. You should not be able to add more than 100 units of stock or an error will be given.  *else if (addStock > 100) {*  *System.out.println("Error: Cannot Add Less Than 1 Unit of Stock")* | **Printing item stock information:**  . . .  Units of Stock To Add: 101  Increasing Stock By 101 Units  Error: Cannot Add More Than 100 Units of Stock |
| **Add < 1 Unit of Stock** | Checks for valid input of adding stock. You should not be able to add less than 1 unit of stock or an error will be given.  *if (addStock < 1) {*  *System.out.println("Error: Cannot Add Less Than 1 Unit of Stock")* | **Printing item stock information:**  . . .  Units of Stock To Add: 0  Increasing Stock By 0 Units  Error: Cannot Add Less Than 1 Unit of Stock |
| **Add Between 1 and 100 Units of Stock** | Checks for valid input of adding stock. The units added should be between 1 and 100 inclusive therefore the lower bounds of 1 and the upper bounds of 100 should be tested as well as numbers in-between.  There should not be any errors returned and the new stock information should add the input to the “Total Units In Stock” | **Printing item stock information:**  . . .  Total Units In Stock: 10  Units of Stock To Add: 1  Increasing Stock By 1 Units  Printing item stock information:  . . .  Total Units In Stock: 11  **Printing item stock information:**  . . .  Total Units In Stock: 10  Units of Stock To Add: 100  Increasing Stock By 100 Units  Printing item stock information:  . . .  Total Units In Stock: 110  **Printing item stock information:**  . . .  Total Units In Stock: 10  Units of Stock To Add: 50  Increasing Stock By 50 Units  Printing item stock information:  . . .  Total Units In Stock: 60 |
| **Sell Stock < 1 Units** | Checks for valid input of selling stock. The units sold cannot be less than 1 unit of stock.  if (sellStock < 1) {  System.out.println("Error: Cannot Sell Less Than 1 Unit of Stock") | **Printing item stock information:**  . . .  Units of Stock Sold: 0  Sold 0 Units of Stock  Error: Cannot Sell Less Than 1 Unit of Stock |
| **Sell Stock Should Not Exceed Available Units of Stock** | Checks for valid input of selling stock. The units sold cannot exceed the available units of stock available. | **Printing item stock information:**  . . .  Total Units In Stock: 10  Units of Stock Sold: 11  Sold 11 Units of Stock  Error: Cannot Sell More Than Available Units of Stock |
| **VAT is 17.5% (double vat = 17.5)** | Calculates the price of the price inclusive of the VAT at the set percentage of 17.5 which is 1.175x price excluding VAT shown in the code below.  getPrice() \* (1 + (getVAT() / 100) | Price (excl. VAT): £99.99  Price (inc. VAT): £117.48825 |
| **toString() method** | Returns string giving stock code, stock name, description, quantity, price before VAT, price after VAT. | **Printing item stock information:**  Stock Type: Unknown Stock Name  Stock Description: Unknown Stock Description  Stock Code: W101  Price (excl. VAT): £99.99  Price (inc. VAT): £117.48825  Total Units In Stock: 10 |

**TASK 2.2**

|  |  |  |
| --- | --- | --- |
| **TEST CASE** | **PURPOSE** | **EXPECTED RESULTS** |
| **Initiation of Stock with Parameters:**  **(“String stockCode, int quantity, double price”)**  **Override the instance methods** | Initiating a new stock:  StockItem stock = new NavSys("NS102", 10, 124.99);  The instance methods in StockItem will be overridden with the NavSys classes getStockName() and getStockDescription() | **Printing item stock information:**  Stock Type: Navigation System  Stock Description: GeoVision Sat Nav  Stock Code: NS102  Price (excl. VAT): £124.99  Price (inc. VAT): £146.86325  Total Units In Stock: 10 |

**TASK 3.2**

|  |  |  |
| --- | --- | --- |
| **TEST CASE** | **PURPOSE** | **EXPECTED RESULT** |
| **Test Polymorphism** | Build an array containing one instance of each of the subclasses of StockItem given the parameter.  The program should run through the array and call the class methods to test each instance.  *s[0] = new NavSys("NS102", 10, 99.99);*  *s[1] = new DashCam("DC302", 15, 150);*  *s[2] = new CarStereo("CS105", 5, 50);*  *s[3] = new WiperBlades("WB203", 20, 30);* | **Printing item stock information:**  Stock Type: Navigation System  Stock Description: GeoVision Sat Nav  Stock Code: NS102  Price (excl. VAT): £99.99  Price (inc. VAT): £117.48825  Total Units In Stock: 10  . . .  **Printing item stock information:**  Stock Type: Dash Cam  Stock Description: Nextbase Dash Cam  Stock Code: DC302  Price (excl. VAT): £150.0  Price (inc. VAT): £176.25  Total Units In Stock: 15  . . .  **Printing item stock information:**  Stock Type: Car Stereo  Stock Description: JVC Car Stereo  Stock Code: CS105  Price (excl. VAT): £50.0  Price (inc. VAT): £58.75  Total Units In Stock: 5  . . .  **Printing item stock information:**  Stock Type: Wiper Blades  Stock Description: Bosch Wiper Blades  Stock Code: WB203  Price (excl. VAT): £30.0  Price (inc. VAT): £35.25  Total Units In Stock: 20  . . . |