Amy Wang, Junhao Shen Project 1

|  |  |  |  |
| --- | --- | --- | --- |
| Test # | Purpose | Input Data | Expected output |
| 1 | Test the add method to see if a GroceryItem object is added to the ShoppingBag array. Then printed out the shopping bag to see items. | new GroceryItem("bread", 3.97, false);  new GroceryItem("apples", 2.84, false); | ·bread: $3.97 : tax free  ·apples: $2.84 : tax free |
| 2 | Test the salesTax method to see if it correctly calculates the amount of tax for items in the shopping cart.   * Case 1: No taxable items, returns $0.00 * Case 2: With taxable item,   returns amount of tax | * Case 1:   ·bread: $3.97 : tax free  ·apples: $2.84 : tax free   * Case 2:   ·bread: $3.97 : tax free  ·apples: $2.84 : tax free  ·napkins: $4.50 : is taxable | * Case 1: $0.00 * Case 2: $0.30 |
| 3 | Test the remove method to see if the given item is successfully removed from the grocery bag. The contents of the shopping bag is printed out before and after to see if the items are correct and in proper order. The following cases are tested consecutively.   * Case 1: Remove from beginning of bag, return true and the last item should replace the first item and last item becomes null * Case 2: Remove from end of bag, return true * Case 3: Remove item not in bag, return false and bag not changed | * Case 1:   ·bread: $3.97 : tax free  ·apples: $2.84 : tax free  ·napkins: $4.50 : is taxable  testbag.remove(new GroceryItem("bread", 3.97, false);)   * Case 2:   ·napkins: $4.50 : is taxable  ·apples: $2.84 : tax free  testbag.remove(new GroceryItem("apples", 2.84, false);   * Case 3:   ·napkins: $4.50 : is taxable  testbag.remove(new GroceryItem("bread", 3.97, false);) | * Case 1: true   ·napkins: $4.50 : is taxable  ·apples: $2.84 : tax free   * Case 2: true   ·napkins: $4.50 : is taxable   * Case 3: false   ·napkins: $4.50 : is taxable |
| 4 | Test grow method to see if capacity of the shopping bag and size of array increases, and if the contents of the shopping bag successfully transfers.   * Case 1: added 6 items to bag with capacity of 5 to force bag to grow, return 10 and print contents * Case 2: directly tested method, after case 1, return capacity 15 | * Case 1:   testbag.add(new GroceryItem("bread", 3.97, false);)  // add command called 6 times   * Case 2:   testbag.grow(); | * Case 1: 10   ·bread: $3.97 : tax free  ·bread: $3.97 : tax free  ·bread: $3.97 : tax free  ·bread: $3.97 : tax free  ·bread: $3.97 : tax free  ·bread: $3.97 : tax free   * Case 2: 15 |