Required Methods - Line-by-Line Frequency Analysis

Products.SearchProductById(int id)

Statement	Freq	Total
if (products.empty()) return null;	1	1
products.findfirst();	1	1
while (true) {	n+1 (condition check)	n+1
if (products.retrieve().getProductId() == id) return products.retrieve();	n	n
if (products.last()) break;	n	n
products.findenext();	n-1	n-1
}	0	0
return null;	1	1

Products.displayOutOfStock()

Statement	Freq	Total
System.out.println("Out of stock products:");	1	1
if (products.empty())	1	1
System.out.println("no products exist");	1 (if empty)	1
else {	0	0
boolean found = false;	1	1
products.findfirst();	1	1
while (true) {	n+1	n+1
if (products.retrieve().getStock() == 0)	n	n
System.out.println(products.retrieve().toString());	<= n	<= n
found = true;	<= n	<= n
if (products.last()) break;	n	n

products.findenext();	n-1	n-1
}	0	0
if (!found) System.out.println("All products in stock");	1	1
}	0	0

Products.addProduct(Product p)

Statement	Freq	Total
if (SearchProductById(p.getProductId()) == null) {	1	n
products.addLast(p);	1	1
System.out.println("Product added: " + p.getName());	1	1
saveAll();	1	n
} else {	0	0
System.out.println("Product with ID ... already exists!");	1 (if exists)	1
}	0	0

Products.removeProduct(int id)

Statement	Freq	Total
if (products.empty()) { System.out.println("Product ID not found"); return; }	1	1
products.findfirst();	1	1
while (true) {	n+1	n+1
if (products.retrieve().getProductId() == id) {	n	n
products.remove();	1 (if found)	1
System.out.println("Product removed: " + id);	1 (if found)	1

saveAll();	1 (if found)	n
return;	1 (if found)	1
}	0	0
if (products.last()) break;	n	n
products.findenext();	n-1	n-1
}	0	0
System.out.println("Product ID not found");	1 (if not found)	1

Products.updateProduct(int id, Product p)

Statement	Freq	Total
Product old = SearchProductById(id);	1	n
if (old == null)	1	1
System.out.println("not exist to make update");	1 (if not found)	1
else {	0	0
old.UpdateProduct(p);	1	1
saveAll();	1	n
}	0	0

Customers.addCustomer(Customer c)

Statement	Freq	Total
if (searchById(c.getCustomerId()) == null) {	1	n
customers.addLast(c);	1	1
System.out.println(" Added customer: " + c.getName());	1	1
saveAll();	1	n
} else {	0	0
System.out.println(" Customer with ID ... already exists!");	1	1

}	0	0

Customer.displayOrders()

Statement	Freq	Total
if (orders.empty()) {	1	1
System.out.println("No orders for customer " + name);	1 (if empty)	1
return;	1 (if empty)	1
}	0	0
System.out.println("Orders for " + name + ":");	1	1
orders.findfirst();	1	1
while (true) {	k+1	k+1
System.out.println(orders.retrieve().toString());	n	n
if (orders.last()) break;	n	n
orders.findenext();	k-1	k-1
}	0	0

Orders.searchOrderById(int id)

Statement	Freq	Total
if (all_orders.empty()) return null;	1	1
all_orders.findfirst();	1	1
while (true) {	n+1	n+1
Order o = all_orders.retrieve();	n	n
if (o.getOrderId() == id) return o;	n	n
if (all_orders.last()) break;	n	n
all_orders.findenext();	n-1	n-1
}	0	0
return null;	1	1

Orders.removeOrderById(int id)

Statement	Freq	Total
if (all_orders.empty()) { System.out.println("Order ID not found"); return; }	1	1
all_orders.findfirst();	1	1
while (true) {	n+1	n+1
if (all_orders.retrieve().getOrderId() == id) {	n	n
all_orders.remove();	1 (if found)	1
System.out.println("Order removed: " + id);	1 (if found)	1
saveAll();	1 (if found)	n
return;	1 (if found)	1
}	0	0
if (all_orders.last()) break;	n	n
all_orders.findenext();	n-1	n-1
}	0	0
System.out.println("Order ID not found");	1 (if not found)	1

Orders.addOrder(Order ord)

Statement	Freq	Total
if (searchOrderById(ord.getOrderId()) == null) {	1	n
all_orders.addLast(ord);	1	1
assign(ord);	1	n
saveAll();	1	n
} else {	0	0
System.out.println("Order with ID ... already exists!");	1	1

}	0	0

Order.setStatus(String status)

Statement	Freq	Total
this.status = status;	1	1

Reviews.addReview(Review r)

Statement	Freq	Total
if (SearchReviewById(r.getReviewID()) == null) {	1	n
reviews.addLast(r);	1	1
assign_to_product(r);	1	n
assign_to_customer(r);	1	n
System.out.println("Review added: " + r.getReviewID());	1	1
saveAll();	1	n
} else {	0	0
System.out.println("Review with ID ... already exists");	1	1
}	0	0

Reviews.updateReview(int id, Review p)

Statement	Freq	Total
Review old = SearchReviewById(id);	1	n
if (old == null)	1	1
System.out.println("not exist to make update");	1 (if not found)	1
else {	0	0
old.UpdateReview(p);	1	1
saveAll();	1	n

}	0	0

Product.getAverageRating()

Statement	Freq	Total
if (reviews.empty()) return 0.0;	1	1
reviews.findfirst();	1	1
double sum = 0;	1	1
int count = 0;	1	1
while (true) {	n+1	n+1
sum += reviews.retrieve().getRating();	n	n
count++;	n	n
if (reviews.last()) break;	n	n
reviews.findenext();	n-1	n-1
}	0	0
return sum / count;	1	1