

ORDER SERVICE SCENARIO

In Figure 1 the activity *GetCart* specifying the behavior of the operation *OrderService.getCart()* is depicted. If a given customer is associated with an existing cart, that cart is provided as output of the activity. Otherwise, a new cart object is created, set as cart of a given customer, and provided as output of the activity.

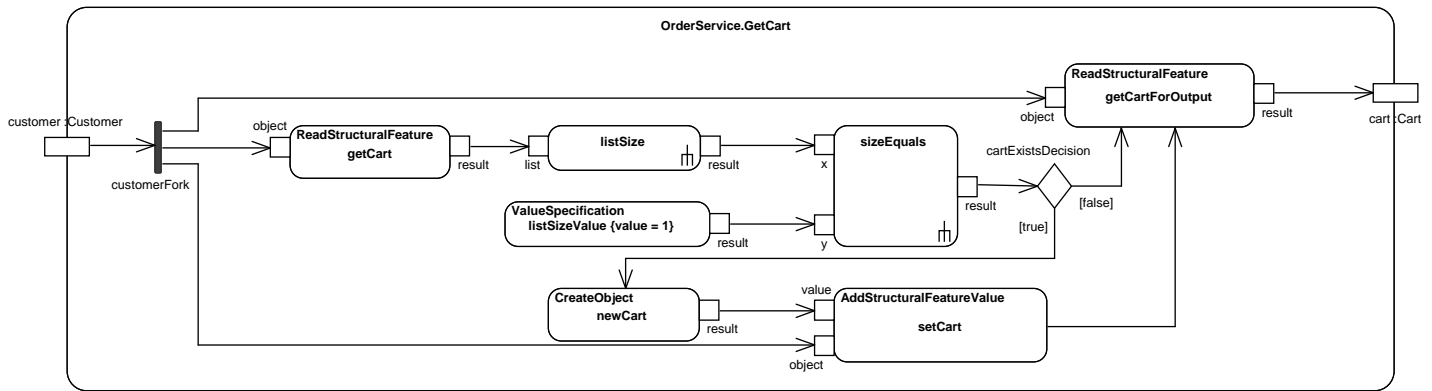


Figure 1: Activity *GetCart* of the *OrderService* class

TEST CASES AND SCENARIOS

Listing 2 and 3 show test cases for evaluating the correctness of the activity. In Listing 4 and 5, the test results are shown indicating that the activity is not correct. Based on the test results, suggest corrections of the activities, such that the tests will be executed successfully.

Listing 1: Test Scenario

```

1 import petstore.data.*
2 import petstore.logic.*
3
4 scenario OrderServiceScenario[
5   object customer: Customer {login='customer'; password='pass'};
6 ]
7
8 scenario OrderServiceScenario2[
9   object customer: Customer{login='customer'; password='pass'};
10  object cart: Cart{}
11
12  link cart_customer{
13    source customer=OrderServiceScenario2.customer;
14    target cart=OrderServiceScenario2.cart;
15  }
16 ]

```

Listing 2: Test Case 1

```

1 test getCartTest activity OrderService.GetCart(customer=OrderServiceScenario.customer){
2   assertOrder *, getCart, *, listSize, *, sizeEquals, cartExistsDecision, newCart, setCart, getCartForOutput;
3   assertState immediately after action getCart{
4     getCart.result = null;
5   }
6   finally{
7     GetCart.cart != null;
8   }
9 }

```

Listing 3: Test Case 2

```

1 test getCartTest2 activity OrderService.GetCart(customer=OrderServiceScenario2.customer){
2   initialize OrderServiceScenario2;
3   assertOrder *, getCart, *, listSize, *, sizeEquals, cartExistsDecision, getCartForOutput;
4   assertState immediately after action getCart{
5     getCart.result = OrderServiceScenario2.cart;
6   }
7   finally{
8     GetCart.cart = OrderServiceScenario2.cart;
9   }
10 }

```

RESULTS OF TEST CASES

Listing 4: Test Case 1 Results Report

```
1 Test Suite Run: 29-09-2014 15:39:41
2
3 TestCase: getCartTest
4 Activity: OrderService.GetCart
5 Activity input: customer = customer;
6   Brute force check:
7   Number of paths checked: 4
8   Number of invalid paths: 4
9
10  Failed path: customerFork, listSizeValue, getCart, listSize, sizeEquals, cartExistsDecision
11  Failed path: customerFork, getCart, listSizeValue, listSize, sizeEquals, cartExistsDecision
12  Failed path: customerFork, getCart, listSize, listSizeValue, sizeEquals, cartExistsDecision
13  Failed path: listSizeValue, customerFork, getCart, listSize, sizeEquals, cartExistsDecision
14
15  Order specification: *, getCart, *, listSize, *, sizeEquals, cartExistsDecision, newCart, setCart, getCartForOutput
16  Matrix validation result: FAIL
17
18  State assertion: IMMEDIATELY AFTER action getCart
19  State expressions checked: 1
20  State expressions failed: 0
21
22  State assertion: ALWAYS AFTER action sizeEquals
23  State expressions checked: 1
24  State expressions failed: 1
25    Expression: GetCart.cart != null / Actual was: NULL
26 *****
```

Listing 5: Test Case 2 Results Report

```
1 Test Suite Run: 29-09-2014 15:39:41
2
3 TestCase: getCartTest2
4 Activity: OrderService.GetCart
5 Activity input: customer = customer;
6   Brute force check:
7   Number of paths checked: 4
8   Number of invalid paths: 4
9
10  Failed path: customerFork, listSizeValue, getCart, listSize, sizeEquals, cartExistsDecision, newCart, setCart
11  Failed path: customerFork, getCart, listSizeValue, listSize, sizeEquals, cartExistsDecision, newCart, setCart
12  Failed path: customerFork, getCart, listSize, listSizeValue, sizeEquals, cartExistsDecision, newCart, setCart
13  Failed path: listSizeValue, customerFork, getCart, listSize, sizeEquals, cartExistsDecision, newCart, setCart
14
15  Order specification: *, getCart, *, listSize, *, sizeEquals, cartExistsDecision, getCartForOutput
16  Matrix validation result: FAIL
17
18  State assertion: IMMEDIATELY AFTER action getCart
19  State expressions checked: 1
20  State expressions failed: 0
21
22  State assertion: ALWAYS AFTER action setCart
23  State expressions checked: 1
24  State expressions failed: 1
25    Expression: GetCart.cart != null / Actual was: NULL
26 *****
```