Exercise 2: Extended data definitions

Please extend the existing rule project EU Rent.

The premium customers of EU Rent are to be saved in the rule project and it should be examined whether the current customer is a premium customer, or not. If so, the premium customer gets a loyalty present.

**Steps:**

1. Create a new input data element on the level of Rental Calculation for the customer ID customer\_ID of type Integer. Define the element as “input” and “required”.
2. Create a new structure Customer on the level of Rental Calculation with the following attributes:

customer\_ID : Integer lastname : String

firstname : String street : String

city : String points : Integer

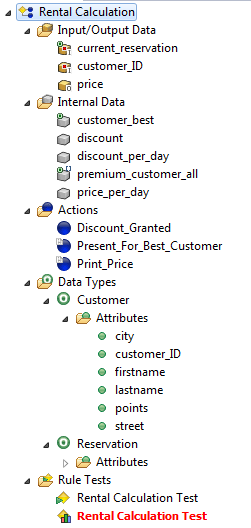
1. Create a new internal data element premium\_customers\_all of type Customer on the level of Rental Calculation and define it as a list.
2. Fill the data element with the following values:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Id | Last name | First name | Street | City | Points |
| 8 | Degas | Edgar | Harvard Street 7 | Los Angeles | 1 |
| 1 | Peters | Carl | Redford Street 12 | Detroit | 25 |
| 3 | Wall | Larry | Long Road 24 | Seattle | 4 |

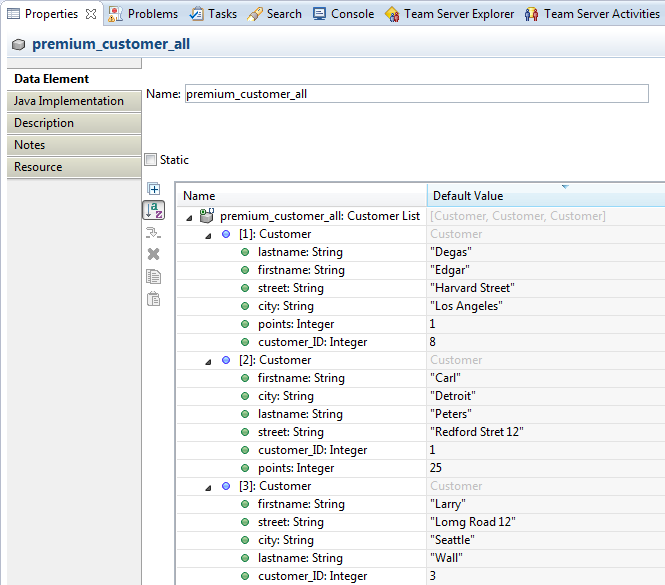
1. Create a new internal data element customers\_best of the type Customer on the level of Rental Calculation.
2. Identify the best customer (sorted by points) of premium\_customers\_all. Save the best customer to data element customers\_best. **Hint**: Use function max(collection: sortExpression) for identifying the best customer.
3. Verify the customer customers\_best by checking, whether his customer ID is equal to the ID of the best customer (defined as input parameter).
4. If applicable, the best customer gets a loyalty present. For this purpose trigger a new action Present\_For\_Best\_Customer and display the following message on the console:„Dear (customer name), you get a present for loyalty!”

Result: Modeling of EU RENT

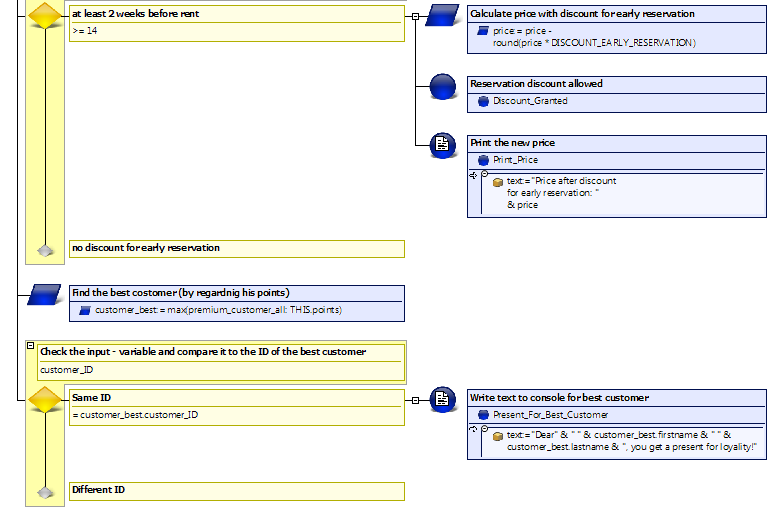
Business data model:



Data element premium\_customers\_all:



Rule Rental Calculation:



Questions:

1. Which of the following states are correct for a list?

A list…

|  |  |  |
| --- | --- | --- |
| a | defines no order |  |
| b | defines an order |  |
| c | supports the access with an index |  |
| d | allows no duplicates |  |
| e | allows duplicates |  |

2. What is the result of the following operations? **Hints**: It is a set.

All\_Customers = All\_CustomersABCD **+** {CustomerE}

A

B

C

D

E

+

:=

New\_Customers = All\_CustomersABCD INTERSECT New\_CustomersABE

A

B

C

D

INTERSECT

:=

A

B

E

B

Results:

1. Which of the following states are correct for a list?

A list…

|  |  |  |
| --- | --- | --- |
| a | defines no order |  |
| b | defines an order | X |
| c | supports the access with an index | X |
| d | allows no duplicates |  |
| e | allows duplicates | X |

2. What is the result of the following operations? **Hints**: It is a set

All\_Customers = All\_CustomersABCD **+** {CustomerE}

A

B

C

D

E

A

B

C

D

E

+

:=

New\_Customers = All\_CustomersABCD INTERSECT New\_CustomersABE

A

B

A

B

C

D

INTERSECT

:=

A

B

E