



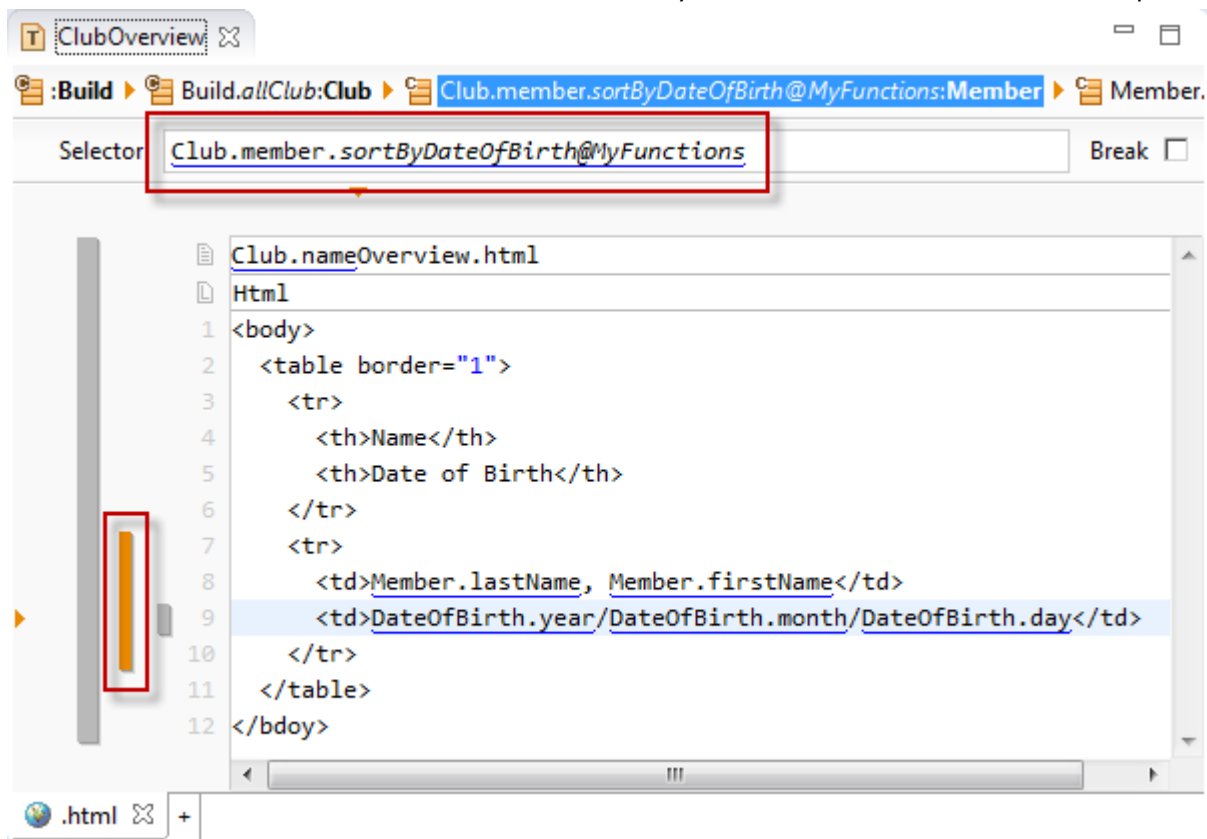


Tutorial

JavaListFunction

Tutorial	Actifsource Tutorial – State Machine
Required Time	<ul style="list-style-type: none"> • 20 Minutes
Prerequisites	<ul style="list-style-type: none"> • Actifsource Tutorial – Installing Actifsource • Actifsource Tutorial – Simple Service • Actifsource Tutorial – Complex Service
Goal	<ul style="list-style-type: none"> • Learn about JavaListFunctions • Create your own JavaListFunction to sort resources
Topics covered	<ul style="list-style-type: none"> • JavaListFunction
Notation	<ul style="list-style-type: none"> •  To do •  Information • Bold: Terms from actifsource or other technologies and tools • <u>Bold underlined</u>: actifsource Resources • <u>Underlined</u>: User Resources • <u><i>UnderlinedItalics</i></u>: Resource Functions • <code>Monospaced</code>: User input • <i>Italics</i>: Important terms in current situation
Disclaimer	<p>The authors do not accept any liability arising out of the application or use of any information or equipment described herein. The information contained within this document is by its very nature incomplete. Therefore, the authors accept no responsibility for the precise accuracy of the documentation contained herein. It should be used rather as a guide and starting point.</p>
Contact	<p>actifsource GmbH Täferstrasse 37 5405 Baden-Dättwil Switzerland www.actifsource.com</p>
Trademark	<p>actifsource is a registered trademark of actifsource GmbH in Switzerland, the EU, USA, and China. Other names appearing on the site may be trademarks of their respective owners.</p>
Compatibility	<p>Created with actifsource Version 6.12.0</p>

- Let's create a simple model for a club with its members with a resource which represents the date of birth
- Let's write a JavalistFunction which can sort members by their date of birth and use it in a template



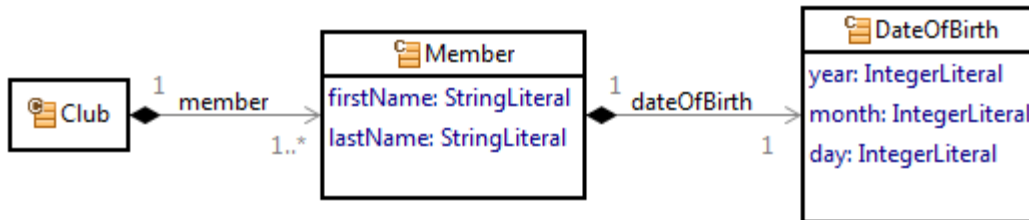
- Create some html output for the club members sorted by the date of birth (month and day)

Name	Date of Birth
Bond, James	1973/02/25
Johnson, James	1935/03/16
Powers, Austin	1943/06/06
Garcia, Maria	2001/07/03
Smith, James	1965/12/04

Part I:

Preparation

4



- ↗ Create a simple design consist of a club, containing members having a date of birth
- ↗ Club is a **NamedResource**
- ↗ Club owns 1..N Member
- ↗ Member is a **Resource**
- ↗ Member contains the following **Attributes** of type **StringLiteral**:
 - firstName
 - lastName
- ↗ Member owns 1_1 DateOfBirth
- ↗ DateOfBirth is a **Resource**
- ↗ DateOfBirth contains the following **Attributes** of type **IntegerLiteral**:
 - Year
 - Month
 - Day

typeOf name comment extends extends modifier aspect[InitializationAspect] aspect[ResourceValidationAspect] aspect[NameAspect]	ch.actifsource.core.Class Member ch.actifsource.core.Resource												
aspect[NameAspect] aspect[NameAspect] + property[1] + property[2] + property[3] property definesAspect allowRoot classIcon lineColor fillColor shape	<table border="1"> <tr> <td>typeOf</td> <td>TextSelectorAspectImplementation</td> </tr> <tr> <td>implements</td> <td>NameAspect</td> </tr> <tr> <td>selector</td> <td><u>Member.lastName</u>, <u>Member.firstName</u></td> </tr> </table> <table border="1"> <tr> <td>firstName</td> <td>: Attribute</td> </tr> <tr> <td>lastName</td> <td>: Attribute</td> </tr> <tr> <td>dateOfBirth</td> <td>: OwnRelation</td> </tr> </table>	typeOf	TextSelectorAspectImplementation	implements	NameAspect	selector	<u>Member.lastName</u> , <u>Member.firstName</u>	firstName	: Attribute	lastName	: Attribute	dateOfBirth	: OwnRelation
typeOf	TextSelectorAspectImplementation												
implements	NameAspect												
selector	<u>Member.lastName</u> , <u>Member.firstName</u>												
firstName	: Attribute												
lastName	: Attribute												
dateOfBirth	: OwnRelation												

- ↳ Create your own name aspect for Member
- ① Note that this step is not necessary for sorting

typeOf	ch.actifsource.core.Class
name	DateOfBirth
comment	
extends	ch.actifsource.core.Resource
extends	
modifier	
aspect[InitializationAspect]	
aspect[ResourceValidationAspect]	
aspect[NameAspect]	
aspect[NameAspect]	
property[1]	
property[2]	
property[3]	
property	
definesAspect	
allowRoot	
classIcon	
lineColor	
fillColor	
shape	

typeOf	TextSelectorAspectImplementation
implements	NameAspect
selector	<u>DateOfBirth.year</u> / <u>DateOfBirth.month</u> / <u>DateOfBirth.day</u>

year	: Attribute
month	: Attribute
day	: Attribute

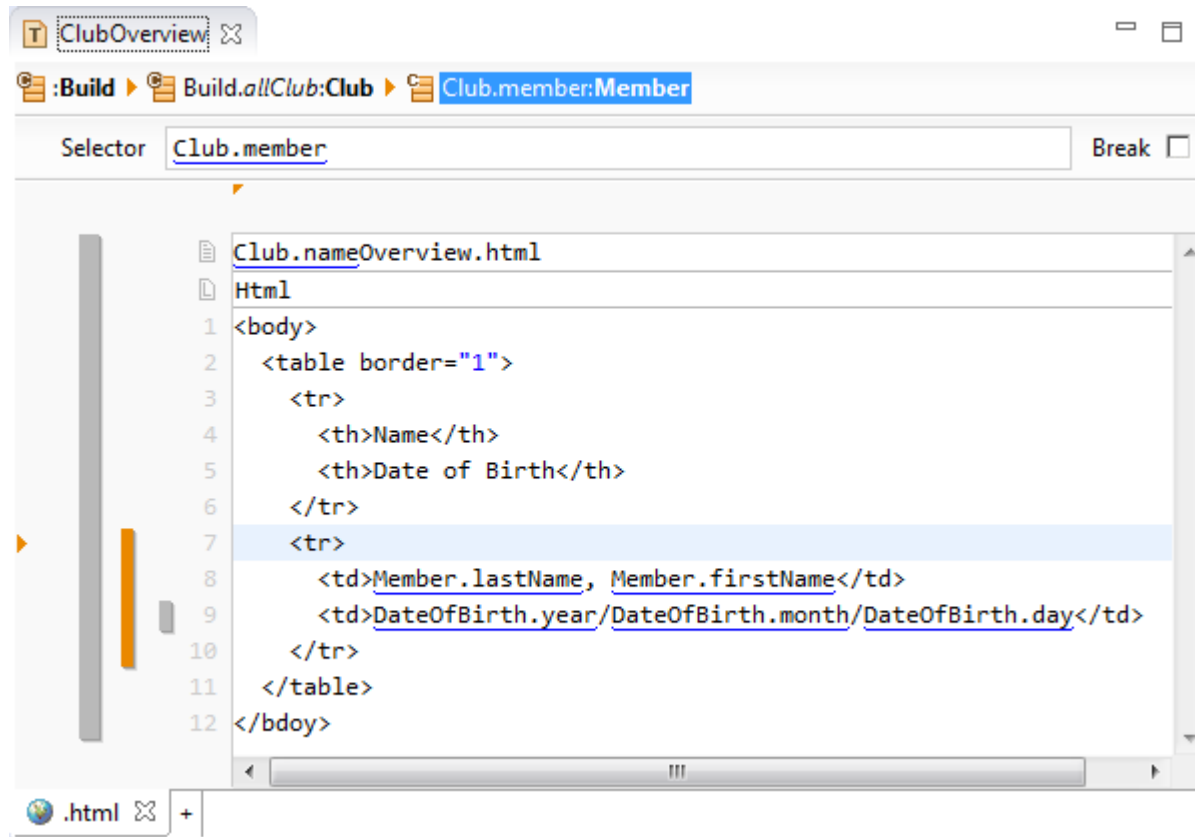
👉 Create your own name aspect for DateOfBirth

📌 Note that this step is not necessary for sorting

Part II:

Creating an HTML Template

7



The screenshot shows an IDE window titled "ClubOverview" with a breadcrumb path: `:Build > Build.allClub:Club > Club.member:Member`. The "Selector" field contains `Club.member`. The main editor displays the following HTML code:

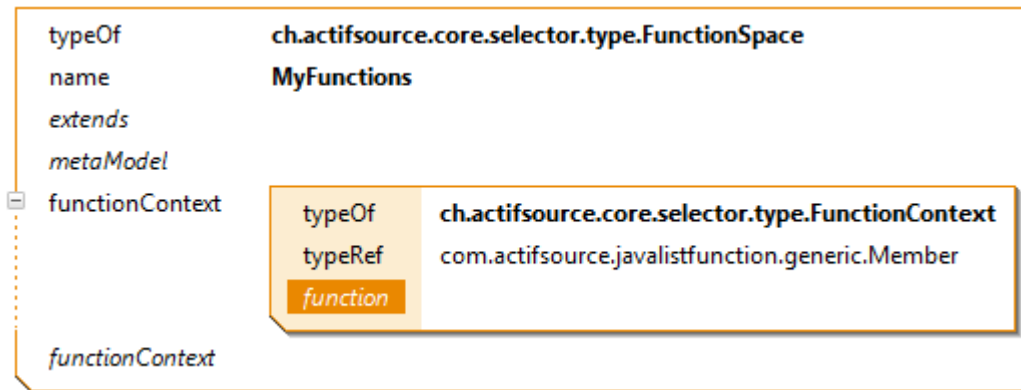
```
Club.nameOverview.html
Html
1 <body>
2   <table border="1">
3     <tr>
4       <th>Name</th>
5       <th>Date of Birth</th>
6     </tr>
7     <tr>
8       <td>Member.lastName, Member.firstName</td>
9       <td>DateOfBirth.year/DateOfBirth.month/DateOfBirth.day</td>
10    </tr>
11  </table>
12 </body>
```

↳ Create a simple html template to display all club members and their birthdays

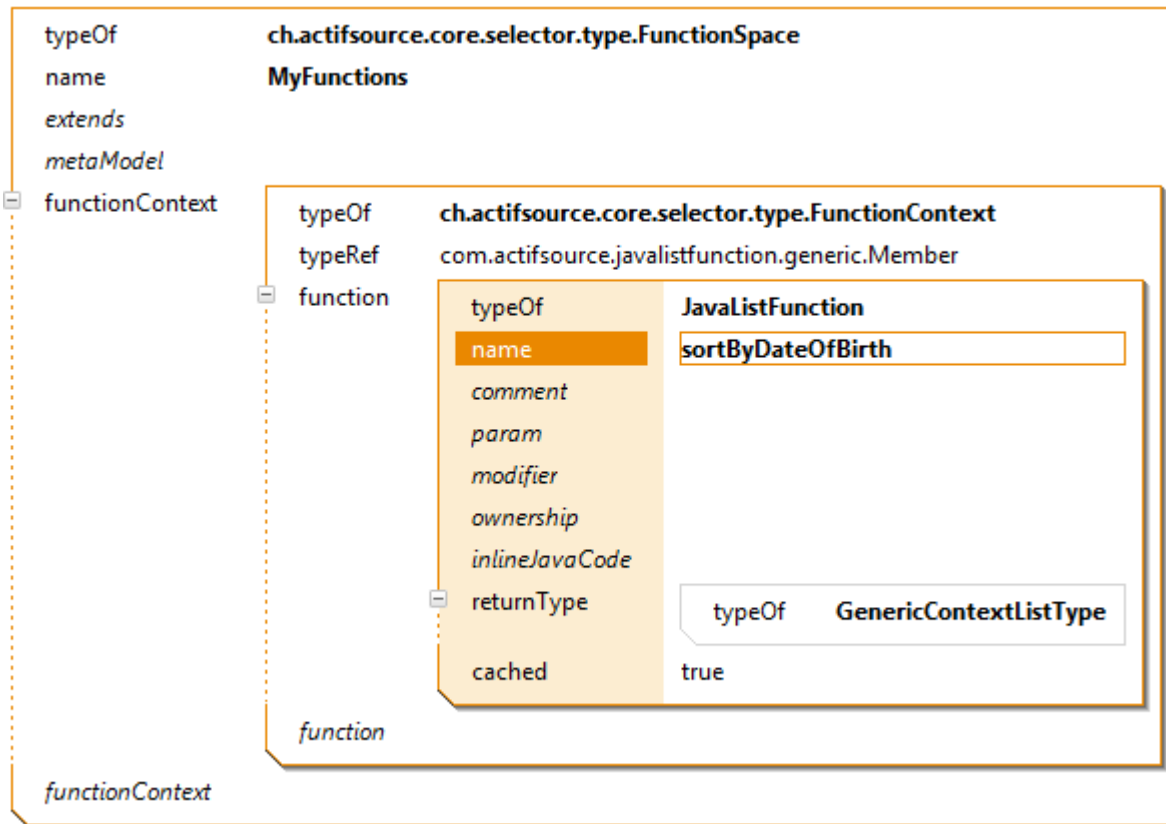
Name	Date of Birth
Smith, James	1965/12/04
Garcia, Maria	2001/07/03
Johnson, James	1935/03/16
Bond, James	1973/02/25
Powers, Austin	1943/06/06

- ① The generated result should look like shown above

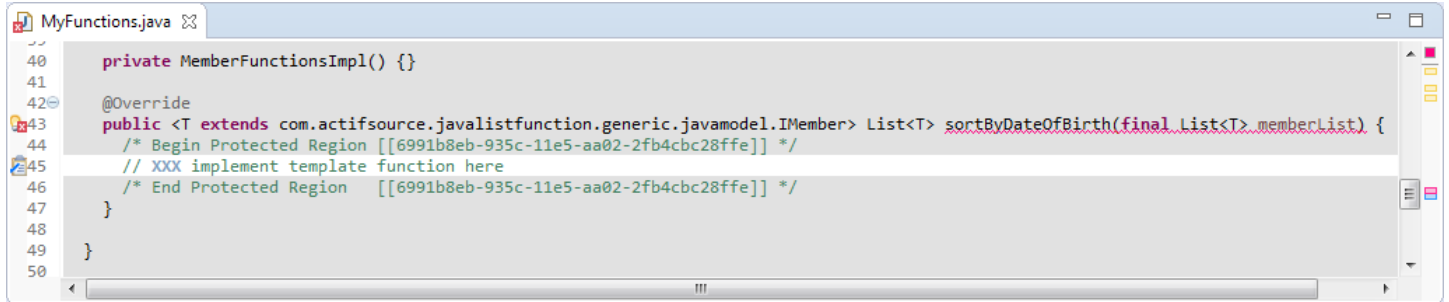
Creating a JavaListFunction



- ↪ Create a new **FunctionSpace** called `MyFunctions`
- ↪ Create a new **FunctionContext** for the **typeRef** `Member`



- ↪ Create a new **JavaListFunction** called *sortByDateOfBirth*
- ↪ Select **GenericContextListType** as **returnType**
- ⓘ Please note that the **GenericContextListType** automatically converts to the same type as used as the function target in the selector
- ⓘ You will love this feature if your type (**typeRef**) has sub classes
- ⓘ Imagine to create a *sortByName* function for **NamedResource**
- ⓘ Calling *A.sortByName* for A as a sub class of **NamedResource** will return A as the type of result and not **NamedResource** even *sortByName* has been defined for **NamedResource**



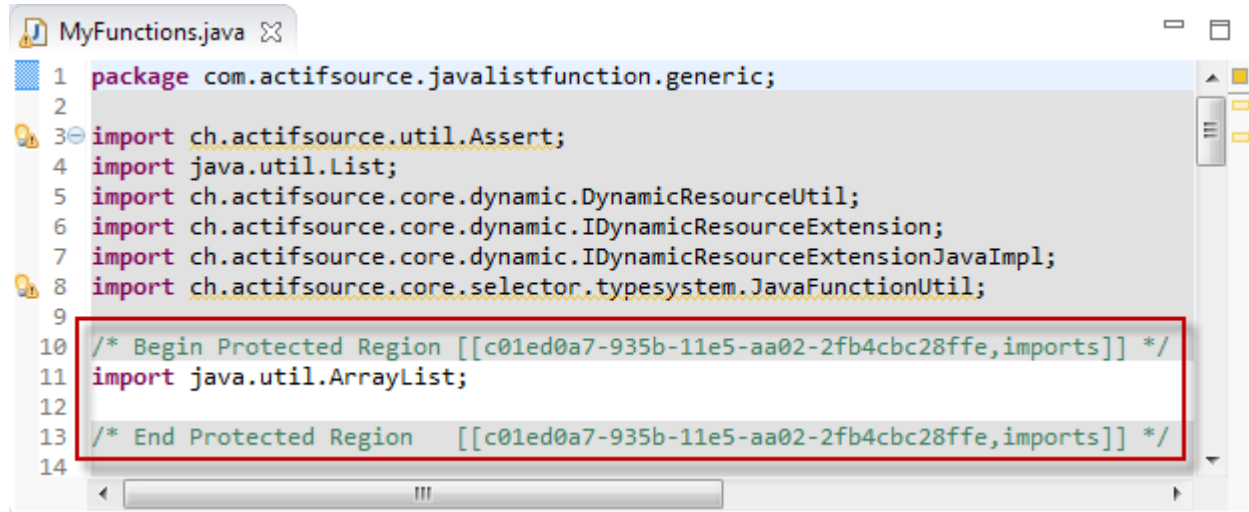
```
MyFunctions.java
40 private MemberFunctionsImpl() {}
41
42 @Override
43 public <T extends com.actifsource.javalistfunction.generic.javamodel.IMember> List<T> sortByDateOfBirth(final List<T> memberList) {
44     /* Begin Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
45     // XXX implement template function here
46     /* End Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
47 }
48
49 }
50
```

- ↪ After saving the **FunctionSpace** MyFunctions the folder *src-gen* will contain a Java file named the same as your FunctionSpace
- ↪ Find the generated function body for the JavaListFunction *sortByDateOfBirth*
- ↪ Let's write some Java code to sort our club members by their date of birth

```
@Override
public <T extends com.actifsource.javalistfunction.generic.javamodel.IMember> List<T> sortByDateOfBirth(final List<T> memberList) {
    /* Begin Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
    ArrayList<T> memberListSorted = new ArrayList<>(memberList);

    return memberListSorted;
    /* End Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
}
```

- ↪ Since the memberList passed by Actifsource is marked final we have to copy the list in order to modify
 - ArrayList<T> memberListSorted = **new** ArrayList<>(memberList);
- ↪ Return the copied list to get rid of compile errors
 - **return** memberListSorted;



```
1 package com.actifsource.javalistfunction.generic;
2
3 import ch.actifsource.util.Assert;
4 import java.util.List;
5 import ch.actifsource.core.dynamic.DynamicResourceUtil;
6 import ch.actifsource.core.dynamic.IDynamicResourceExtension;
7 import ch.actifsource.core.dynamic.IDynamicResourceExtensionJavaImpl;
8 import ch.actifsource.core.selector.typesystem.JavaFunctionUtil;
9
10 /* Begin Protected Region [[c01ed0a7-935b-11e5-aa02-2fb4cbc28ffe,imports]] */
11 import java.util.ArrayList;
12
13 /* End Protected Region [[c01ed0a7-935b-11e5-aa02-2fb4cbc28ffe,imports]] */
14
```

- ① Please make sure to place the import statement for ArrayList in the ProtectedRegion
- ① Otherwise the your import will be overwritten from the generator

```

@Override
public <T extends com.actifsource.javalistfunction.generic.javamodel.IMember> List<T>
sortByDateOfBirth(final List<T> memberList) {
    /* Begin Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
    ArrayList<T> memberListSorted = new ArrayList<>(memberList);

    Collections.sort(memberListSorted, new Comparator<T>() {

        @Override
        public int compare(T o1, T o2) {
            IDateOfBirth dateOfBirth1 = o1.selectDateOfBirth();
            IDateOfBirth dateOfBirth2 = o2.selectDateOfBirth();

            int monthComp = dateOfBirth1.selectMonth().compareTo(dateOfBirth2.selectMonth());
            if (monthComp!=0) {return monthComp;}

            int dayComp = dateOfBirth1.selectDay().compareTo(dateOfBirth2.selectDay());
            if (dayComp!=0) {return dayComp;}

            return 0;
        }
    });

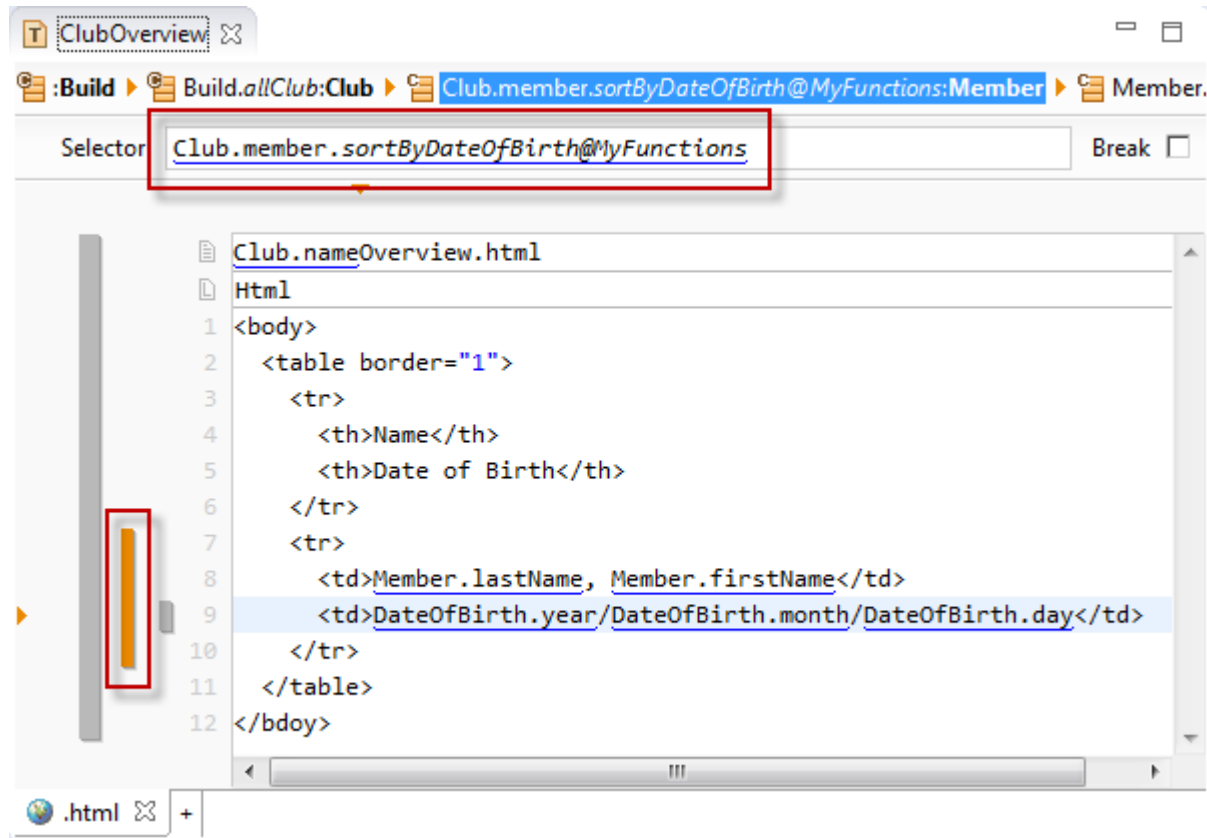
    return memberListSorted;
    /* End Protected Region [[6991b8eb-935c-11e5-aa02-2fb4cbc28ffe]] */
}

```

- ↪ Using the static function Collections.sort() from the Java collection framework allows us to pass a so called Comparator
- ↪ Let's implement the anonymous class of type Comparator<T> as shown above
- ① Comparator.compare shall return 0 if objects are equal, -1 if o1 is a predecessor of o2, or +1 if o1 is a successor of o2
- ① Since the variables o1 and o2 are fully typed we can directly access the properties of Member using the selectXYZ() functions
- ① The code shown above is sorting dates regarding month and day only
- ① Try to implement a sort function which shows the oldest member first

Part IV: Putting it all together

15



The screenshot shows an IDE window titled "ClubOverview". The breadcrumb navigation at the top indicates the current context: `:Build > Build.allClub:Club > Club.member.sortByDateOfBirth@MyFunctions:Member > Member.`. Below this, a "Selector" window is open, containing the text `Club.member.sortByDateOfBirth@MyFunctions`, which is highlighted with a red box. The main code editor displays the following HTML code:

```
1 <body>
2   <table border="1">
3     <tr>
4       <th>Name</th>
5       <th>Date of Birth</th>
6     </tr>
7     <tr>
8       <td>Member.lastName, Member.firstName</td>
9       <td>DateOfBirth.year/DateOfBirth.month/DateOfBirth.day</td>
10    </tr>
11  </table>
12 </body>
```

The code editor also features a vertical scrollbar on the left, which is highlighted with a red box. The status bar at the bottom left shows ".html" and a plus sign.

↪ Use the `sortByDateOfBirth` function to sort any selected list of members by their date of birth

Name	Date of Birth
Bond, James	1973/02/25
Johnson, James	1935/03/16
Powers, Austin	1943/06/06
Garcia, Maria	2001/07/03
Smith, James	1965/12/04

↳ As a result your members are sorted by month and day

