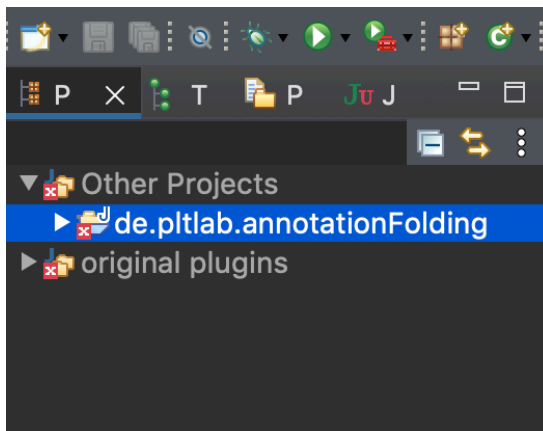
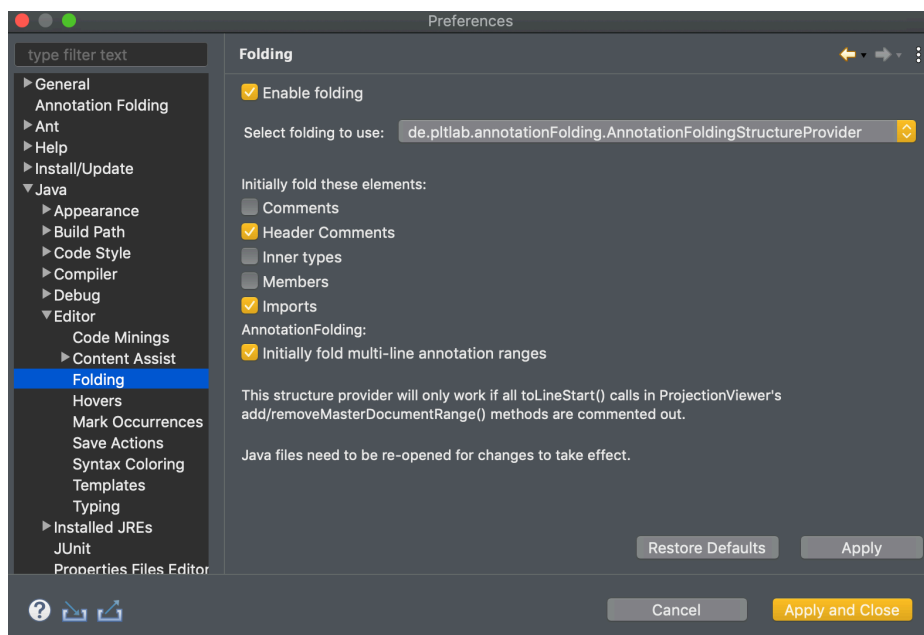


Workspaces.zip contains the development workspace and runtime application workspace for testing.

1. The plug-in project is in workspace_dev (development workspace):



2. Under **Preferences > Java > Editor > Folding** and select:
 - **BasicAnnotationFoldingStructureProvider** for basic annotation folding.
 - **AnnotationFoldingStructureProvider** for the experimental strategy (needs adjusted ProjectionViewer in the workspace to work)



3. runtime-EclipseApplication contains some ready-made test cases:

Basic Folding: (single range strategy)

before folding:

```
77
78 // Case 6
79
80 @Pre("n > 10 && s != null")
81 @Post("n = 10 && s != null")
82 @LongAnnotation(
83     name = "happy_path",
84     requires = {
85         @requires("length > 0 and the bag contains n elements of elem, w=
86         @requires("new cardinality(elem) = old cardinality(elem) - 1"),
87         @requires("returns true") },
88     ensures = {
89         @ensures("length is old length minus 1") })
90 public void peeler3(Fruit fruit) {
91     fruit.peel();
92 }
```

after folding:

```
77
78 // Case 6
79
80 @Pre("n > 10 && s != null")
90 public void peeler3(Fruit fruit) {
91     fruit.peel();
92 }
```

Experimental Strategy: (needs adjusted ProjectionViewer in workspace)

before folding:

```
71 // Case 6
72
73 @Pre ("n > 10 && s != null")
74 @Post("n = 10 && s != null")
75 @LongAnnotation(name = "happy_path",
76     requires = {
77         @requires("length > 0 and the bag contains n elements of ele
78         @requires("new cardinality(elem) = old cardinality(elem) - 1
79         @requires("returns true") },
80     ensures = {
81         @ensures("length is old length minus 1") })
82 public void peeler3(Fruit fruit) {
83     fruit.peel();
84 }
```

after folding:

```
71 // Case 6
72
73 @Pre ()
74 @Post()
75 @LongAnnotation(
82 public void peeler3(Fruit fruit) {
83     fruit.peel();
84 }
```