```
/* (c) Andrzej Wasowski 2014
 * A set of constraints for the FiniteStateMachine example, implemented in
Xtend.
 * I avoid EMF integration for constraints on purpose, to keep this simple.
 * IMHO, you should always avoid the EMF integration for constraints, if you do
 * not explicitly need it.  It seems just way to complex.
 */

package JUnitDomain.validation

import JUnitDomain.Fixture
import JUnitDomain.Suite
import org.eclipse.emf.ecore.EObject
import JUnitDomain.After
import JUnitDomain.Before
import JUnitDomain.Method

class Constraints {

      /* To solve the exercise, just replace the "true" constants
       * with suitable expressions and test them with some instance data */

      def static dispatch constraint (Fixture it) {
            // in every fixture, an after method has a suitable After annotation

            //true
            (after == null || after.annotation instanceof After)
            &&


            // every before method has to be annotated with a Before annotation

            //true
            (before != null || before.annotation instanceof Before)

      }

      def static dispatch constraint (Suite it) {
            // the collection of tests have to be a subset of the inherited
collection of methods

            //true
            tests.forall[t | t instanceof Method]
      }

      /* Fall back for all types that are not constrained */
      def static dispatch boolean constraint(EObject it) {
            true /* do not replace this one :) */
      }



}
```