

```

1 package validation

3 import Configurator.BinaryConstraint
import Configurator.BinaryOperator
5 import Configurator.BooleanLiteral
import Configurator.DoubleLiteral
7 import Configurator.IntLiteral
import Configurator.Literal
9 import Configurator.Parameter
import Configurator.ParameterIdentifier
11 import Configurator.StringLiteral
import Configurator.UnaryConstraint
13 import org.eclipse.emf.ecore.EObject

15 class Constraints {

17     // Constraint operator constraint
    def static dispatch boolean constraint(BinaryConstraint it) {
19         (andOperatorConstraint(it)) || (mathOperatorConstraint(it)
        )
    }

21     def static boolean mathOperatorConstraint(BinaryConstraint it)
    {
23         (operator.equals(BinaryOperator.NOTEQUALS) || operator.
        equals(BinaryOperator.EQUALS) || operator.equals(
        BinaryOperator.GT) || operator.equals(BinaryOperator.GTEQ) ||
        operator.equals(BinaryOperator.LT) || operator.equals(
        BinaryOperator.LTEQ))
        &&
25         (leftOperand instanceof ParameterIdentifier || leftOperand
        instanceof Literal)
        &&
27         (it.rightOperand instanceof ParameterIdentifier || it.
        rightOperand instanceof Literal)
    }

29     def static boolean andOperatorConstraint(BinaryConstraint it
    ) {
31         (operator.equals(BinaryOperator.AND) || operator.equals(
        BinaryOperator.OR) || operator.equals(BinaryOperator.XOR))
        &&
33         (leftOperand instanceof BinaryConstraint || leftOperand
        instanceof UnaryConstraint)
        &&
35         (rightOperand instanceof BinaryConstraint || rightOperand
        instanceof UnaryConstraint)
    }
}

```

```

37 // Parameter can only have either literal or enum value
39 def static dispatch boolean constraint(Parameter it) {
    (literalValues == null && enumValue != null &&
    enumCountConstraint)
41 ||
    (literalValues != null && enumValue == null &&
    literalTypesConstraint && literalCountConstraint)
43 }

45 def static boolean literalCountConstraint(Parameter it) {
    (literalValues.size <= maxChosenValues) && (literalValues.
    size >= minChosenValues)
47 }

49 def static boolean enumCountConstraint(Parameter it) {
    (enumValue.size <= maxChosenValues) && (enumValue.size >=
    minChosenValues)
51 }

53 // Literal values need to be of the same type.
55 def static boolean literalTypesConstraint(Parameter it) {
    literalValues.forall[x | x instanceof IntLiteral]
    ||
57 literalValues.forall[x | x instanceof StringLiteral]
    ||
59 literalValues.forall[x | x instanceof DoubleLiteral]
    ||
61 literalValues.forall[x | x instanceof BooleanLiteral]
    }

63 // Unary constraint can only contains binary constraint
65 def static dispatch boolean constraint(UnaryConstraint it) {
    operand instanceof BinaryConstraint
67 }

69 // Catch all case for dynamic dispatch resolution
71 def static dispatch boolean constraint(EObject it) {
    true
    }
73 }

```

