

Fortress - Task #5421

Utility methods on expression nodes

11/12/2014 02:14 PM - Artem Kotsynyak

Status:	Open	Start date:	11/12/2014
Priority:	Normal	Due date:	
Assignee:	Andrei Tatarnikov	% Done:	20%
Category:	Expression	Estimated time:	0.00 hour
Target version:	0.4	Published in build:	
Detected in build:	svn		
Description			
<ul style="list-style-type: none">• Shortcuts for standard operations, e.g. AND (...) to replace new NodeOperation(StandardOperation.AND, ...) etc.• Default TRUE and FALSE nodes w/o user data.• Node hierarchy checks: check if Node is a specific operation (Node x opId -> boolean), check if Node is a specific DataType value (Node x DataType -> boolean).• Complementary to NodeValue.newXXX: isXXX and getXXX.• Operands extraction. <p>Some of these can be found implemented or even duplicated in the codebase. This list is a subject to updates.</p>			

History

#1 - 11/17/2014 04:24 PM - Andrei Tatarnikov

Для реализации "Shortcuts for standard operations" нужно осудить:

1. Добавление фабричных методов ExprUtils. Там уже есть методы getConjunction, getDisjunction, getNegation, в которые добавлены проверки isCondition. Не нарушат ли новые методы концепцию ExprUtils?
2. Создать новый утилитный класс для статических фабричных методов. Например, OperationFactory (имя не должно быть длинным). Тогда StandardOperations нужно переименовать (класс Operation уже есть в калькуляторе - будет конфликт). Нужно другое имя.

#2 - 12/04/2014 03:15 PM - Andrei Tatarnikov

- Assignee set to Andrei Tatarnikov
- Target version set to 0.3

We decided that, to solve this task, static factory methods should be added to the StandardOperations enumeration (they will have the same name as corresponding enumeration elements).

#3 - 01/26/2015 01:39 PM - Andrei Tatarnikov

- Status changed from New to Open
- % Done changed from 0 to 20

Complementary to NodeValue.newXXX: isXXX and getXXX.

The following methods have been added (r769):

- To the *Data* class:

```
public boolean isType(DataTypeId typeId);
public boolean isType(DataType type);
public BigInteger getInteger();
public BitVector getBitVector();
public boolean getBoolean();
public double getReal();
public DataMap getArray();
```

- To the *Node* class:

```
public boolean isType(DataTypeId typeId);
public boolean isType(DataType type);
```

- To the *NodeValue* class:

```
public BigInteger getInteger();
public BitVector getBitVector();
public boolean getBoolean();
public double getReal();
public DataMap getArray()
```

#4 - 02/08/2015 12:47 PM - Artem Kotsynyak

Nominating method `List<Node> NodeOperation.getOperands()` returning immutable list view on operands.

#5 - 02/09/2015 08:52 AM - Andrei Tatarnikov

Nominating method `List<Node> NodeOperation.getOperands()` returning immutable list view on operands

Done in r782

#6 - 02/09/2015 01:22 PM - Andrei Tatarnikov

Default TRUE and FALSE nodes w/o user data.

Decided not to implement because *NodeValue* is mutable (the `setData` method can change its state).

#7 - 03/25/2015 07:33 PM - Alexander Kamkin

- Subject changed from *[expression] Utility methods on expression nodes* to *Utility methods on expression nodes*
- Category set to *Expression*

#8 - 07/30/2015 06:33 PM - Andrei Tatarnikov

- Target version changed from 0.3 to 0.4

#9 - 04/22/2020 10:13 PM - Alexander Kamkin

Close?