PMD 6.15.0 released

26 May 2019

26-May-2019 - 6.15.0

The PMD team is pleased to announce PMD 6.15.0.

This is a minor release.

Table Of Contents

New and noteworthy

Enhanced Matlab support

Thanks to the contributions from Maikel Steneker CPD for Matlab can now parse Matlab programs which use the question mark operator to specify access to class members:

```
classdef Class1
properties (SetAccess = ?Class2)
```

CPD also understands now double quoted strings, which are supported since version R2017a of Matlab:

```
str = "This is a string"
```

Enhanced C++ support

Example: `auto integer_literal = 1'000'000;`

The single quotes can be used to add some structure to large numbers.

CPD also parses raw string literals now correctly (see #1784).

New Rules

- The new Apex rule `FieldNamingConventions` (`apex-codestyle`) checks the naming conventions for field declarations. By default this rule uses the standard Apex naming convention (Camel case), but it can be configured through properties.
- The new Apex rule `FormalParameterNamingConventions` (`apex-codestyle`) checks the naming conventions for formal parameters of methods. By default this rule uses the standard Apex naming convention (Camel case), but it can be configured through properties.
- The new Apex rule `LocalVariableNamingConventions` (`apex-codestyle`) checks the naming conventions for local variable declarations. By default this rule uses the standard Apex naming convention (Camel case), but it can be configured through properties.
- The new Apex rule `PropertyNamingConventions` (`apex-codestyle`) checks the naming conventions for property declarations. By default this rule uses the standard Apex naming convention (Camel case), but it can be configured through properties.
- The new Java rule `UseShortArrayInitializer` (`java-codestyle`) searches for array initialization expressions, which can be written shorter.

Modified Rules

- The Apex rule `ClassNamingConventions` (`apex-codestyle`) can now be configured using various properties for the specific kind of type declarations (e.g. class, interface, enum). As before, this

- The Apex rule `MethodNamingConventions` (`apex-codestyle`) can now be configured using various properties to differenciate e.g. static methods and test methods. As before, this rule uses by default the standard Apex naming convention (Camel case).

- The Java rule `FieldNamingConventions` (`java-codestyle`) now by default ignores the field `serialPersistentFields`. Since this is a field which needs to have this special name, no field naming conventions can be applied here. It is excluded the same way like `serialVersionUID` via the property `exclusions`.

- The Java rule `CommentRequired` (`java-documentation`) has a new property `serialPersistentFieldsCommentRequired` with the default value "Ignored". This means that from now on comments for the field `serialPersistentFields` are not required anymore. You can change the property to restore the old behavior.

- The Java rule `ProperLogger` (`java-errorprone`) has two new properties to configure the logger class (e.g. "org.slf4j.Logger") and the logger name of the special case, when the logger is not static. The name of the static logger variable was already configurable. The new property "loggerClass" allows to use this rule for different logging frameworks. This rule covers all the cases of the now deprecated rule `LoggerIsNotStaticFinal`.

- The Java rule `CommentDefaultAccessModifier` (`java-codestyle`) now reports also missing comments for top-level classes and annotations, that are package-private.

Deprecated Rules

- The Apex rule `VariableNamingConventions` (`apex-codestyle`) has been deprecated and will be removed with PMD 7.0.0. The rule is replaced by the more general rules `FieldNamingConventions`, `FormalParameterNamingConventions`, `LocalVariableNamingConventions`, and `PropertyNamingConventions`.

is replaced by `ProperLogger`.

Fixed Issues

- apex
  - #1321: [apex] Should VariableNamingConventions require properties to start with a lowercase letter?
  - #1783: [apex] comments on constructor not recognized when the Class has inner class
- cpp
  - #1784: [cpp] Improve support for raw string literals
- dart
  - #1809: [dart] [cpd] Parse error with escape sequences
- java
  - #1842: [java] Annotated module declarations cause parse error
- java-bestpractices
  - #1738: [java] MethodReturnsInternalArray does not work in inner classes
- java-codestyle
  - #1495: [java] Rule to detect overly verbose array initializiation
  - #1684: [java] Properly whitelist serialPersistentFields
  - #1804: [java] NPE in UnnecessaryLocalBeforeReturnRule
- python
  - #1810: [python] [cpd] Parse error when using Python 2 backticks
- matlab
  - #1830: [matlab] [cpd] Parse error with comments
  - #1793: [java] CommentDefaultAccessModifier not working for classes

API Changes

Deprecated APIs

For removal

- The `DumpFacades` in all languages, that could be used to transform a AST into a textual representation, will be removed with PMD 7. The rule designer is a better way to inspect nodes.

    - net.sourceforge.pmd.lang.ecmascript.ast.DumpFacade
    - net.sourceforge.pmd.lang.jsp.ast.DumpFacade
    - net.sourceforge.pmd.lang.plsql.ast.DumpFacade
    - net.sourceforge.pmd.lang.vf.ast.DumpFacade
    - net.sourceforge.pmd.lang.vm.ast.AbstractVmNode#dump
    - net.sourceforge.pmd.lang.xml.ast.DumpFacade
- The method LanguageVersionHandler#getDumpFacade will be removed as well. It is deprecated, along with all its implementations in the subclasses of LanguageVersionHandler.

External Contributions

- #1647: [java] Rule to detect overly verbose array initialization - Victor
- #1762: [java] LoggerIsNotStaticFinal and ProperLogger - make class-name configurable - Ivo Šmíd
- #1798: [java] Make CommentDefaultAccessModifier work for top-level classes - Boris Petrov
- #1799: [java] MethodReturnsInternalArray does not work in inner classes - Fixed #1738 - Srinivasan Venkatachalam
- #1802: [python] [cpd] Add support for Python 2 backticks - Maikel Steneker
- #1803: [dart] [cpd] Dart escape sequences - Maikel Steneker
- #1807: [ci] Fix missing local branch issues when executing pmd-regression-tester - BBG
- #1813: [matlab] [cpd] Matlab comments - Maikel Steneker
- #1816: [apex] Fix ApexDoc handling with inner classes - Jeff Hube
- #1817: [apex] Add configurable naming convention rules - Jeff Hube
- #1819: [cpp] [cpd] Add support for digit separators - Maikel Steneker
- #1820: [cpp] [cpd] Improve support for raw string literals - Maikel Steneker
- #1821: [matlab] [cpd] Matlab question mark token - Maikel Steneker
- #1822: [matlab] [cpd] Double quoted string - Maikel Steneker
- #1837: [core] Minor performance improvements - Michael Hausegger

- #1840: [Java] Whitelist serialPersistentFields - Marcel Härle