Projet CIR2

BURGER QUIZZ

Qualimétrie

michael.aron@isen.fr
yann.le-ru@isen.fr

Mis à jour le 05/06/2015

Notation (prévisionnelle)

- Commentaires :

Pourcentage	Note
<5%	0%
Entre 5 et 10%	50%
>10%	100%

- Duplication

Pourcentage	Note
<5%	100%
Entre 5 et 10%	75%
Entre 10 et 15%	50%
Entre 15 et 20%	25%
>20%	0%

Notation (prévisionnelle)

- Respect des règles (major & minor):

Pourcentage	Note
80%	0%
…	…
90%	50%
…	…
100%	100%

- Violation des règles critiques ou bloquantes : une erreur => 0/20 pour la qualité du code

PHP : règles bloquantes

Blocker — A goto-statement is used — Phppmd_rules

Goto makes code harder to read and it is nearly impossible to understand the control flow of an application that uses this language construct.
Therefore it should be avoided. Consider to replace Goto with regular control structures and separate methods/function, which are easier to read.

Example:

```
class Foo {
    public function bar($param)  {
        A:
        if ($param === 42) {
            goto X;
        }
        Y:
        if (time() % 42 === 23) {
            goto Z;
        }
        X:
        if (time() % 23 === 42) {
            goto Y;
        }
        Z:
        return 42;
    }
}
```

Key: Design Rules/GotoStatement

Blocker — DuplicateClassNameFound — Php_codesniffer_rules

Duplicate %s name "%s" found; first defined in %s on line %s

Key: Generic.Classes.DuplicateClassName.Found

Blocker — FunctionDuplicateArgumentFound — Php_codesniffer_rules

Variable "%s" appears more than once in function declaration

Key: Squiz.Functions.FunctionDuplicateArgument.Found

Blocker — StaticThisUsageFound — Php_codesniffer_rules

Usage of "$this" in static methods will cause runtime errors

Key: Squiz.Scope.StaticThisUsage.Found

PHP : règles critiques

PHP : règles majeures 1

Major — A class has too many children. — Phppmd_rules

A class with an excessive number of children is an indicator for an unbalanced class hierarchy. You should consider to refactor this class hierarchy.

minimum: Maximum number of acceptable child classes. Default is 12.
Key: Design Rules/NumberOfChildren

Major — A class has too many parents. — Phppmd_rules

A class with many parents is an indicator for an unbalanced and wrong class hierarchy. You should consider to refactor this class hierarchy.

minimum: Maximum number of acceptable parent classes. Default is 6.
Key: Design Rules/DepthOfInheritance

Major — An aval-expression is used — Phppmd_rules

An eval-expression is untestable, a security risk and bad practice. Therefore it should be avoided. Consider to replace the eval-expression with regular code.

Example:

```
class Foo {
    public function bar($param)  {
        if ($param === 42) {
            eval('$param = 23;');
        }
    }
}
```

Key: Design Rules/EvalExpression

Major — An exit-expression is used — Phppmd_rules

An exit-expression within regular code is untestable and therefore it should be avoided. Consider to move the exit-expression into some kind of startup script where an error/exception code is returned to the calling environment.

Example:

```
class Foo {
    public function bar($param)  {
        if ($param === 42) {
            exit(23);
        }
    }
}
```

Key: Design Rules/ExitExpression

Major — ConstructorNameOldStyle — Php_codesniffer_rules

PHP4 style constructors are not allowed; use "__construct()" instead

Key: Generic.NamingConventions.ConstructorName.OldStyle

PHP : règles majeures 2

Major	EmptyStatementNotAllowed Empty %s statement detected Key: Generic.CodeAnalysis.EmptyStatement.NotAllowed	Php_codesniffer_rules
Major	GlobalKeywordNotAllowed Use of the "global" keyword is forbidden; use "$GLOBALS[\'%s\']" instead Key: Squiz.PHP.GlobalKeyword.NotAllowed	Php_codesniffer_rules
Major	NonExecutableCodeReturnNotRequired Empty return statement not required here Key: Squiz.PHP.NonExecutableCode.ReturnNotRequired	Php_codesniffer_rules

Major	Number of class fields exceeds maximum
	Classes that have too many fields could be redesigned to have fewer fields, possibly through some nested object grouping of some of the information. For example, a class with city/state/zip fields could instead have one Address field.
	Example:
	`class Person {`
	`  protected $one;`
	`  private $_two;`
	`  private $_three;`
	`  [... many more fields ...]`
	`}`
	maxfields: The field count reporting threshold. Default is 15.
	Key: Code Size Rules/TooManyFields
		Phppmd_rules

Major	Number of methods in class exceeds maximum A class with too many methods is probably a good suspect for refactoring, in order to reduce its complexity and find a way to have more fine grained objects. maxmethods: The method count reporting threshold. Default is 10. Key: Code Size Rules/TooManyMethods	Phppmd_rules
Major	ObjectInstantiationNotAssigned New objects must be assigned to a variable Key: Squiz.Objects.ObjectInstantiation.NotAssigned	Php_codesniffer_rules
Major	UnconditionalIfStatementFound Avoid IF statements that are always true or false Key: Generic.CodeAnalysis.UnconditionalIfStatement.Found	Php_codesniffer_rules

PHP : règles majeures 3

Major — Unused local variable — Phppmd_rules

A local variable is declared and/or assigned, but not used.

Example:

```
class Foo {
    public function doSomething()
    {
        $i = 5; // Unused
    }
}
```

Key: Unused Code Rules/UnusedLocalVariable

Major — Unused private field — Phppmd_rules

A private field is declared and/or assigned a value, but not used.

Example:

```
class Something
{
    private static $FOO = 2; // Unused
    private $i = 5;          // Unused
    private $j = 6;
    public function addOne()
    {
        return $this->j++;
    }
}
```

Key: Unused Code Rules/UnusedPrivateField

Major — Unused private method — Phppmd_rules

A private method is declared but is unused.

Example:

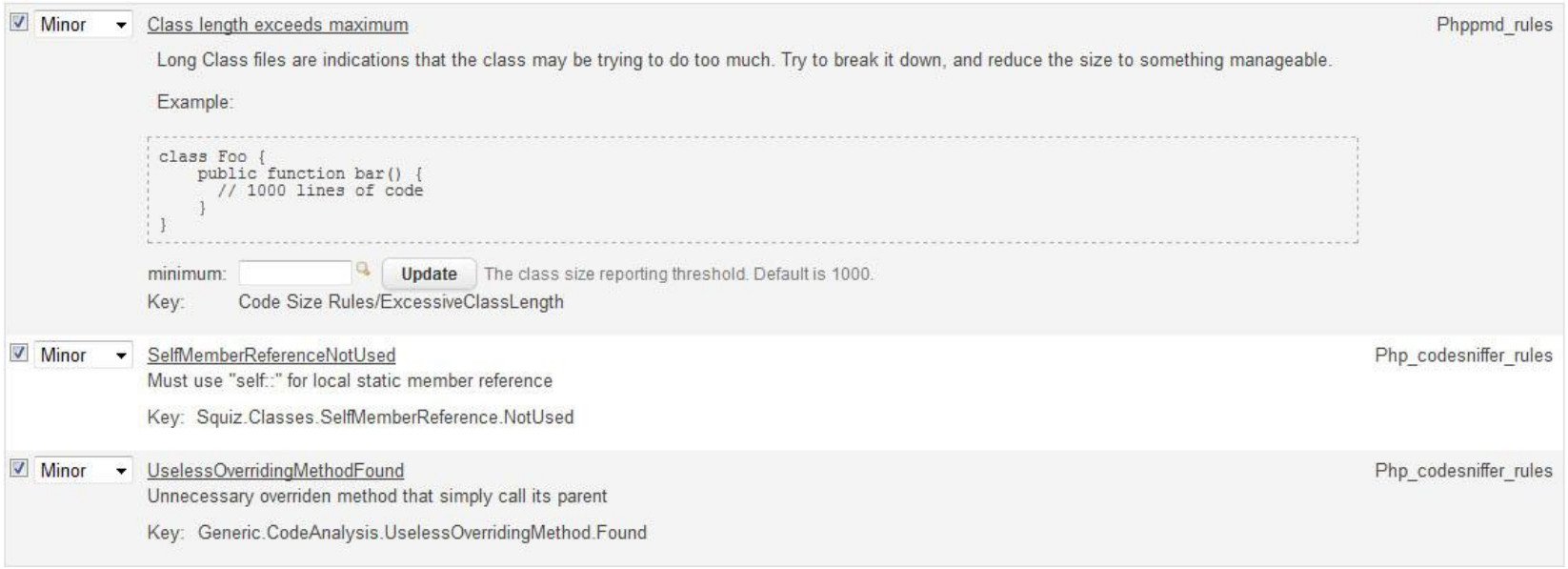
```
class Something
{
    private function foo() {} // unused
}
```

Key: Unused Code Rules/UnusedPrivateMethod

PHP : règles majeures 4

Major	UnusedFunctionParameterFound The method parameter %s is never used Key: Generic.CodeAnalysis.UnusedFunctionParameter.Found	Php_codesniffer_rules
Major	UpperCaseConstantNameConstantNotUpperCase Constants must be uppercase; expected %s but found %s Key: Generic.NamingConventions.UpperCaseConstantName.ConstantNotUpperCase	Php_codesniffer_rules
Major	ValidDefaultValueNotAtEnd Arguments with default values must be at the end of the argument list Key: PEAR.Functions.ValidDefaultValue.NotAtEnd	Php_codesniffer_rules

PHP : règles mineures



PHP : règles infos

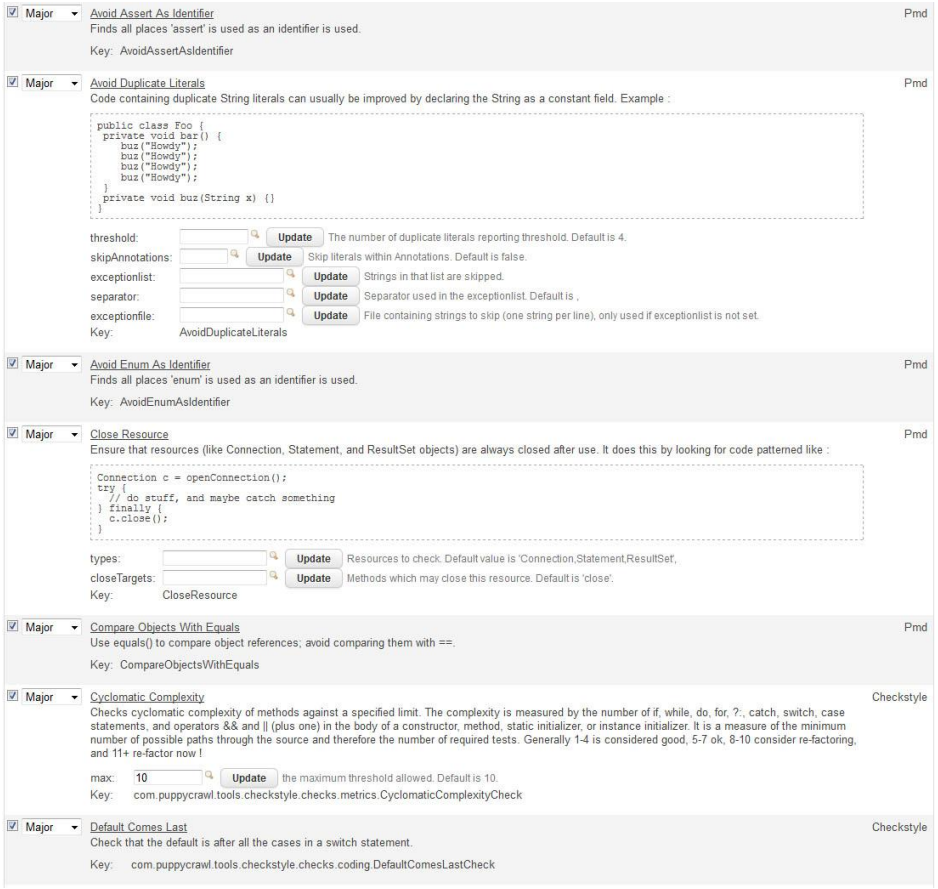
Java : règles critiques

Critical	Empty Finally Block Avoid empty finally blocks - these can be deleted. Key: EmptyFinallyBlock	Pmd
Critical	Empty If Stmt Empty If Statement finds instances where a condition is checked but nothing is done about it. Key: EmptyIfStmt	Pmd
Critical	Empty While Stmt Empty While Statement finds all instances where a while statement does nothing. If it is a timing loop, then you should use Thread.sleep() for it; if it's a while loop that does a lot in the exit expression, rewrite it to make it clearer. Key: EmptyWhileStmt	Pmd
Critical	Equals Null Inexperienced programmers sometimes confuse comparison concepts and use equals() to compare to null. Key: EqualsNull	Pmd

Critical	Naming - Suspicious equals method name
	The method name and parameter number are suspiciously close to equals(Object), which may mean you are intending to override the equals(Object) method.
	Example :
	`public class Foo {`
	`  public int equals(Object o) {`
	`  // oops, this probably was supposed to be boolean equals`
	`  }`
	`  public boolean equals(String s) {`
	`  // oops, this probably was supposed to be equals(Object)`
	`  }`
	`}`
	Key: SuspiciousEqualsMethodName
		Pmd

Critical	Unconditional If Statement Do not use if statements that are always true or always false. Key: UnconditionalIfStatement	Pmd

Java : règles majeures 1



Major — Avoid Assert As Identifier — Pmd
Finds all places 'assert' is used as an identifier is used.
Key: AvoidAssertAsIdentifier

Major — Avoid Duplicate Literals — Pmd
Code containing duplicate String literals can usually be improved by declaring the String as a constant field. Example :

```
public class Foo {
 private void bar() {
     buz("Howdy");
     buz("Howdy");
     buz("Howdy");
     buz("Howdy");
 }
 private void buz(String x) {}
}
```

threshold: Update — The number of duplicate literals reporting threshold. Default is 4.
skipAnnotations: Update — Skip literals within Annotations. Default is false.
exceptionlist: Update — Strings in that list are skipped.
separator: Update — Separator used in the exceptionlist. Default is ,
exceptionfile: Update — File containing strings to skip (one string per line), only used if exceptionlist is not set.
Key: AvoidDuplicateLiterals

Major — Avoid Enum As Identifier — Pmd
Finds all places 'enum' is used as an identifier is used.
Key: AvoidEnumAsIdentifier

Major — Close Resource — Pmd
Ensure that resources (like Connection, Statement, and ResultSet objects) are always closed after use. It does this by looking for code patterned like :

```
Connection c = openConnection();
try {
  // do stuff, and maybe catch something
} finally {
  c.close();
}
```

types: Update — Resources to check. Default value is 'Connection,Statement,ResultSet'.
closeTargets: Update — Methods which may close this resource. Default is 'close'.
Key: CloseResource

Major — Compare Objects With Equals — Pmd
Use equals() to compare object references; avoid comparing them with ==.
Key: CompareObjectsWithEquals

Major — Cyclomatic Complexity — Checkstyle
Checks cyclomatic complexity of methods against a specified limit. The complexity is measured by the number of if, while, do, for, ?:, catch, switch, case statements, and operators && and || (plus one) in the body of a constructor, method, static initializer, or instance initializer. It is a measure of the minimum number of possible paths through the source and therefore the number of required tests. Generally 1-4 is considered good, 5-7 ok, 8-10 consider re-factoring, and 11+ re-factor now !
max: 10 Update — the maximum threshold allowed. Default is 10.
Key: com.puppycrawl.tools.checkstyle.checks.metrics.CyclomaticComplexityCheck

Major — Default Comes Last — Checkstyle
Check that the default is after all the cases in a switch statement.
Key: com.puppycrawl.tools.checkstyle.checks.coding.DefaultComesLastCheck

Java : règles majeures 2

Major — Empty Switch Statements
Avoid empty switch statements.
Key: EmptySwitchStatements

Major — Empty Try Block
Avoid empty try blocks - what's the point?
Key: EmptyTryBlock

Major — Exception As Flow Control
Using Exceptions as flow control leads to GOTOish code and obscures true exceptions when debugging.
Key: ExceptionAsFlowControl

Major — Final Class
Checks that class which has only private constructors is declared as final.
Key: com.puppycrawl.tools.checkstyle.checks.design.FinalClassCheck

Major — Finalize Does Not Call Super Finalize
If the finalize() is implemented, its last action should be to call super.finalize.
Key: FinalizeDoesNotCallSuperFinalize

Major — Finalize Overloaded
Methods named finalize() should not have parameters. It is confusing and probably a bug to overload finalize(). It will not be called by the VM.
Key: FinalizeOverloaded

Major — For Loops Must Use Braces
Avoid using 'for' statements without using curly braces, like `for (int i=0; i<42;i++) foo();`
Key: ForLoopsMustUseBraces

Major — Hidden Field
Checks that a local variable or a parameter does not shadow a field that is defined in the same class.

tokens:	VARIABLE_DEF	Update	tokens to check
ignoreFormat:		Update	pattern for names to ignore
ignoreConstructorParameter:	true	Update	Controls whether to ignore constructor parameters. Default is false.
ignoreSetter:	true	Update	Controls whether to ignore the parameter of a property setter method, where the property setter method for field 'xyz' has nam
ignoreAbstractMethods:	true	Update	Controls whether to ignore parameters of abstract methods. Default is false.

Key: com.puppycrawl.tools.checkstyle.checks.coding.HiddenFieldCheck

Major — Hide Utility Class Constructor
Make sure that utility classes (classes that contain only static methods) do not have a public constructor.
Key: com.puppycrawl.tools.checkstyle.checks.design.HideUtilityClassConstructorCheck

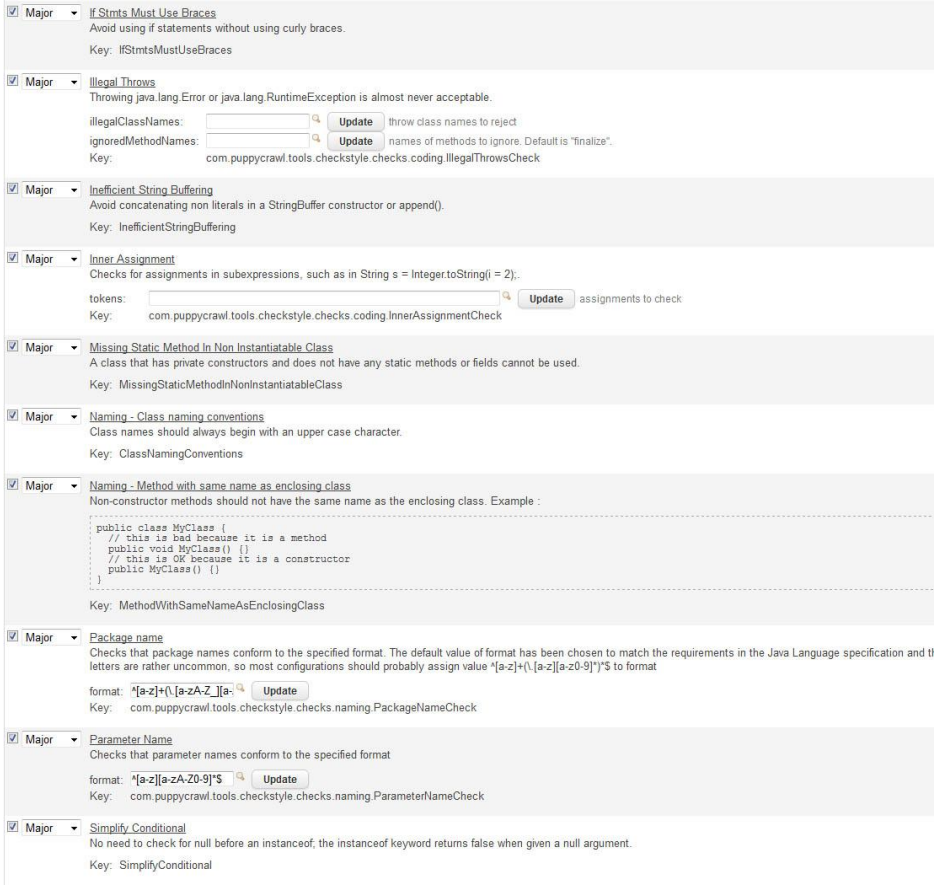
Major — Idempotent Operations
Avoid idempotent operations - they are have no effect. Example :

```
int x = 2;
x = x;
```

Key: IdempotentOperations

Major — If Else Stmts Must Use Braces
Avoid using if..else statements without using curly braces.
Key: IfElseStmtsMustUseBraces

Java : règles majeures 3



Major — If Stmts Must Use Braces
Avoid using if statements without using curly braces.
Key: IfStmtsMustUseBraces

Major — Illegal Throws
Throwing java.lang.Error or java.lang.RuntimeException is almost never acceptable.
illegalClassNames: Update throw class names to reject
ignoredMethodNames: Update names of methods to ignore. Default is "finalize".
Key: com.puppycrawl.tools.checkstyle.checks.coding.IllegalThrowsCheck

Major — Inefficient String Buffering
Avoid concatenating non literals in a StringBuffer constructor or append().
Key: InefficientStringBuffering

Major — Inner Assignment
Checks for assignments in subexpressions, such as in String s = Integer.toString(i = 2);.
tokens: Update assignments to check
Key: com.puppycrawl.tools.checkstyle.checks.coding.InnerAssignmentCheck

Major — Missing Static Method In Non Instantiatable Class
A class that has private constructors and does not have any static methods or fields cannot be used.
Key: MissingStaticMethodInNonInstantiatableClass

Major — Naming - Class naming conventions
Class names should always begin with an upper case character.
Key: ClassNamingConventions

Major — Naming - Method with same name as enclosing class
Non-constructor methods should not have the same name as the enclosing class. Example :

```
public class MyClass {
  // this is bad because it is a method
  public void MyClass() {}
  // this is OK because it is a constructor
  public MyClass() {}
}
```

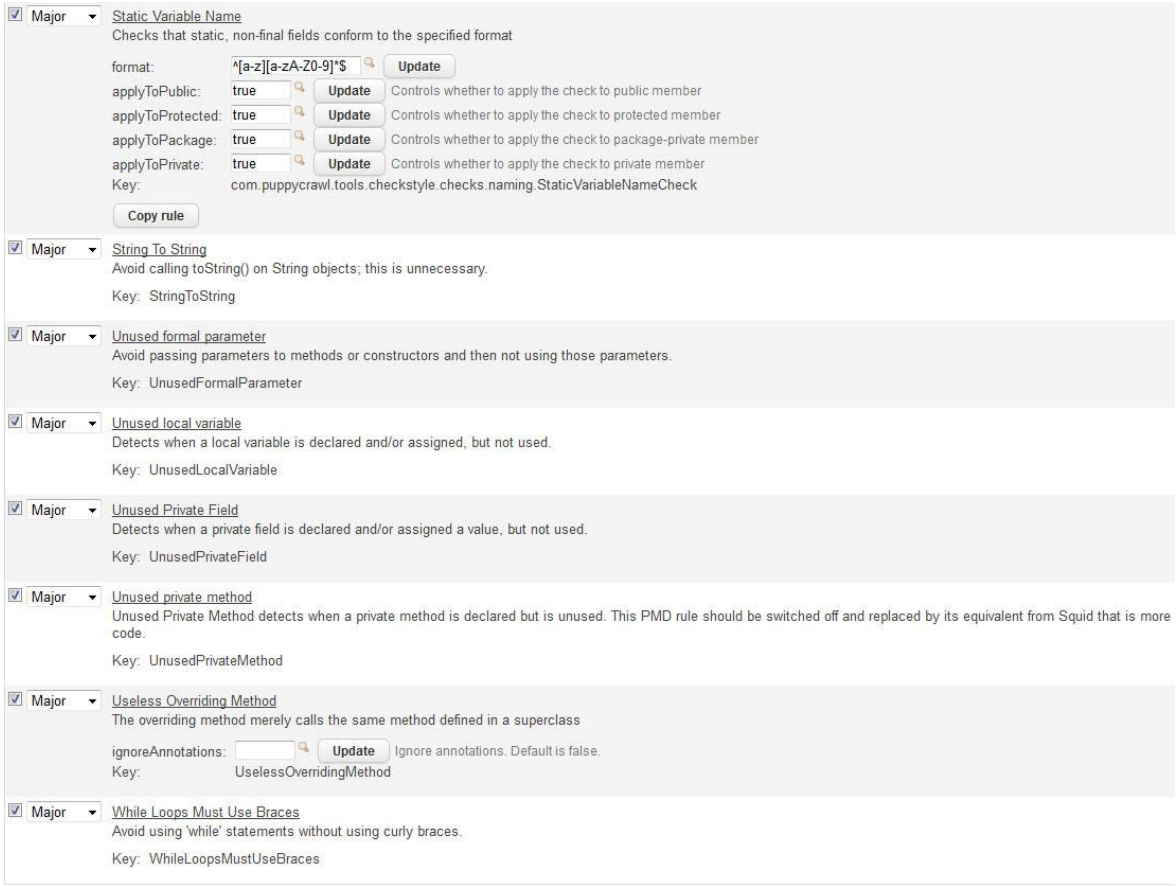
Key: MethodWithSameNameAsEnclosingClass

Major — Package name
Checks that package names conform to the specified format. The default value of format has been chosen to match the requirements in the Java Language specification and th
letters are rather uncommon, so most configurations should probably assign value ^[a-z]+(\.[a-z][a-z0-9]*)*$ to format
format: ^[a-z]+(\.[a-zA-Z_][a- Update
Key: com.puppycrawl.tools.checkstyle.checks.naming.PackageNameCheck

Major — Parameter Name
Checks that parameter names conform to the specified format
format: ^[a-z][a-zA-Z0-9]*$ Update
Key: com.puppycrawl.tools.checkstyle.checks.naming.ParameterNameCheck

Major — Simplify Conditional
No need to check for null before an instanceof; the instanceof keyword returns false when given a null argument.
Key: SimplifyConditional

Java : règles majeures 4



Major — Static Variable Name
Checks that static, non-final fields conform to the specified format

format: ^[a-z][a-zA-Z0-9]*$ Update
applyToPublic: true Update Controls whether to apply the check to public member
applyToProtected: true Update Controls whether to apply the check to protected member
applyToPackage: true Update Controls whether to apply the check to package-private member
applyToPrivate: true Update Controls whether to apply the check to private member
Key: com.puppycrawl.tools.checkstyle.checks.naming.StaticVariableNameCheck
Copy rule

Major — String To String
Avoid calling toString() on String objects; this is unnecessary.
Key: StringToString

Major — Unused formal parameter
Avoid passing parameters to methods or constructors and then not using those parameters.
Key: UnusedFormalParameter

Major — Unused local variable
Detects when a local variable is declared and/or assigned, but not used.
Key: UnusedLocalVariable

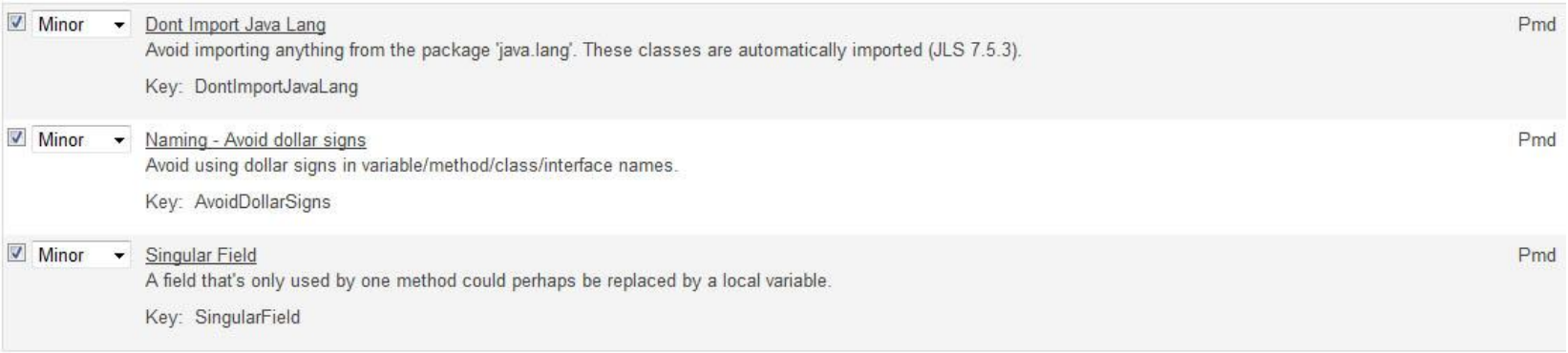
Major — Unused Private Field
Detects when a private field is declared and/or assigned a value, but not used.
Key: UnusedPrivateField

Major — Unused private method
Unused Private Method detects when a private method is declared but is unused. This PMD rule should be switched off and replaced by its equivalent from Squid that is more code.
Key: UnusedPrivateMethod

Major — Useless Overriding Method
The overriding method merely calls the same method defined in a superclass
ignoreAnnotations: Update Ignore annotations. Default is false.
Key: UselessOverridingMethod

Major — While Loops Must Use Braces
Avoid using 'while' statements without using curly braces.
Key: WhileLoopsMustUseBraces

Java : règles mineures



☑	Minor	Dont Import Java Lang Avoid importing anything from the package 'java.lang'. These classes are automatically imported (JLS 7.5.3). Key: DontImportJavaLang	Pmd
☑	Minor	Naming - Avoid dollar signs Avoid using dollar signs in variable/method/class/interface names. Key: AvoidDollarSigns	Pmd
☑	Minor	Singular Field A field that's only used by one method could perhaps be replaced by a local variable. Key: SingularField	Pmd

Java : règles infos

FIN

Des questions?