UTOPIOS
TRANSFORMATION DIGITALE INTELLIGENTE

# JAVA CLEAN CODE Partie 3

# Découverte et utilisation des outils de qualimétrie dans le monde Java

- Introduction frameworks et outils de qualimètrie.
- SonarQube
- Checkstyle, FindBugs
- Cobertura et JaCoCo
- Intégration des outils de qualimétrie dans une PIC.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## Introduction frameworks et outils de qualimètrie.

- La qualimétrie logicielle est le processus d'analyse et d'amélioration de la qualité d'un logiciel.
- Elle inclut la mesure de divers attributs du logiciel, tels que sa fiabilité, sa maintenabilité, sa complexité, etc.
- Dans l'écosystème Java, plusieurs outils et frameworks sont disponibles pour aider à ce processus.
- SonarQube : C'est un outil de qualimétrie de code populaire qui prend en charge de nombreux langages, y compris Java. SonarQube peut analyser le code pour détecter les bugs, code smells, les vulnérabilités de sécurité, et fournir une couverture de code. Il peut également mesurer la dette technique, qui est une estimation du temps nécessaire pour corriger tous les problèmes de code identifiés.
- Checkstyle : Checkstyle est un outil de développement qui aide les programmeurs à écrire du code Java qui adhère à un ensemble de règles de codage. Il automatisera le processus de vérification du code Java pour le rendre plus conforme à certaines conventions de codage.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## Introduction frameworks et outils de qualimètrie.

- PMD (Programming Mistake Detector) : C'est un autre outil d'analyse de code statique qui peut détecter les bugs potentiels, les mauvaises pratiques de codage, les expressions compliquées ou inutiles, les duplications de code, etc.
- FindBugs/SpotBugs : C'est un outil d'analyse statique de bytecode Java qui détecte les bugs potentiels dans le code. SpotBugs est le successeur de FindBugs, offrant des fonctionnalités similaires.
- JaCoCo (Java Code Coverage) : JaCoCo est un outil de couverture de code qui identifie les parties de votre code qui ne sont pas testées. Il est généralement utilisé en conjonction avec des outils de test unitaire comme JUnit.
- JUnit : Bien que ce soit un framework de test unitaire, JUnit joue également un rôle important dans la qualimétrie en permettant aux développeurs d'écrire et d'exécuter des tests pour vérifier que le code se comporte comme prévu

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## SonarQube.

- SonarQube est un outil d'analyse statique de code largement utilisé pour améliorer la qualité du code en identifiant les problèmes potentiels.

1. Analyse de Code : SonarQube peut analyser le code source pour une grande variété de langages de programmation (Java, JavaScript, C#, Python, etc.) pour détecter les problèmes de qualité.
2. Détection des Bugs et Vulnérabilités : Il est capable de détecter une variété de problèmes de qualité, y compris les bugs potentiels et les vulnérabilités de sécurité.
3. Détection des "Code Smells" (Odeurs de Code) : SonarQube peut identifier les "code smells", qui sont des caractéristiques du code qui indiquent une mauvaise conception. Les "code smells" peuvent rendre le code plus difficile à comprendre et à maintenir.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

# SonarQube.

4. Mesure de la Couverture de Tests : En se connectant à des outils de couverture de code comme JaCoCo pour Java ou dotCover pour C#, SonarQube peut mesurer le pourcentage de votre code qui est couvert par des tests.
5. Calcul de la Dette Technique : SonarQube estime le temps qu'il faudrait pour corriger tous les problèmes de qualité qu'il a détectés, une mesure appelée "dette technique". Il affiche également une note de maintenabilité sur une échelle de A à E pour aider à comprendre rapidement la qualité du code.
6. Duplication de Code : SonarQube peut détecter les parties du code qui sont dupliquées, ce qui peut indiquer une mauvaise conception ou un risque accru d'erreurs.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

# SonarQube.

7. Analyse de l'Historique du Code : SonarQube peut analyser l'historique du code pour identifier les tendances dans la qualité du code au fil du temps.
8. Rapports et Tableaux de Bord : SonarQube fournit des rapports détaillés et des tableaux de bord qui peuvent être utilisés pour surveiller la qualité du code à différents niveaux, de l'ensemble de l'entreprise jusqu'au module individuel.
9. Intégration Continue/Déploiement Continu (CI/CD) : SonarQube s'intègre facilement dans les pipelines CI/CD, ce qui permet d'effectuer une analyse de la qualité du code à chaque commit ou avant chaque déploiement.
10. Règles et Profils de Qualité : SonarQube fournit un ensemble de règles par défaut pour chaque langage qu'il supporte, et les utilisateurs peuvent également définir leurs propres règles ou modifier les règles existantes. Ces règles peuvent être regroupées en "profils de qualité" qui peuvent être appliqués à un ou plusieurs projets.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## Checkstyle.

- Checkstyle est un outil de développement qui aide à garantir que votre code Java respecte certaines conventions de codage. Il est très configurable et peut être ajusté pour correspondre à vos propres conventions de codage.
- **Vérification du Style de Code** : Checkstyle peut vérifier que votre code adhère à une variété de conventions de style, comme les règles de formatage, le nommage des variables, l'utilisation des accolades, etc.
- **Vérification de la Conception** : Checkstyle peut détecter les mauvaises pratiques de conception, comme les classes avec trop de responsabilités, les dépendances cycliques, etc.
- **Vérification des commentaires** : Checkstyle peut vérifier la présence et le format des commentaires de documentation Javadoc.
- **Vérification des Importations** : Checkstyle peut vérifier que votre code n'utilise que les importations nécessaires et qu'il ne contient pas d'importations inutilisées.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## Checkstyle.

1. Installation du plugin (maven, gradle,...)
2. Executer le check
3. Consulter le rapport généré en format html, target/site/checkstyle.html

# Découverte et utilisation des outils de qualimétrie dans le monde Java

# PMD.

- PMD est un outil d'analyse de code source pour les langages de programmation tels que Java, JavaScript, XML, et plus encore. Il examine le code pour détecter les mauvaises pratiques potentielles comme les variables non utilisées, les blocs catch vides, les classes inutiles, et plus encore.

1. **Détection de bugs potentiels** : PMD peut identifier les erreurs de programmation courantes comme les variables non initialisées, les exceptions vides, etc.
2. **Amélioration de la lisibilité du code** : Il détecte les "code smells", qui sont des indices de problèmes potentiels dans le code qui peuvent rendre le code plus difficile à lire et à maintenir.
3. **Performance** : PMD détecte les problèmes de performance courants, comme l'utilisation inutile d'objets String ou les boucles inutiles.
4. **Sécurité** : Il détecte les problèmes de sécurité comme l'utilisation de hard-coding, les exceptions silencieuses, etc.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## PMD.

1. Installation du plugin (maven, gradle,...)
2. Executer le check
3. Consulter le rapport généré en format html, target/site/pmd.html

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## FindBugs.

- FindBugs est un outil d'analyse statique de code pour Java qui détecte les erreurs de programmation potentielles. Il utilise l'analyse de bytecode, ce qui signifie qu'il fonctionne sur les fichiers compilés (.class) et non sur le code source directement. FindBugs est capable de trouver une variété de problèmes de qualité de code, y compris :

1. Les erreurs de nullité (par exemple, les références null potentiellement déréférencées).
2. Les violations de la précision des points flottants.
3. Les problèmes de performance, tels que les objets String mal utilisés.
4. Les problèmes de sécurité, comme les variables de classe exposées publiquement.
5. Les mauvaises pratiques de codage, telles que les instructions switch sans default.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

## JaCoCo.

- JaCoCo est un outil populaire de couverture de code pour Java. Il vous permet de voir quelles parties de votre code sont couvertes par vos tests unitaires et quelles parties ne le sont pas. Cela peut être très utile pour identifier les zones de votre code qui pourraient nécessiter des tests supplémentaires.
- Les principales fonctionnalités de JaCoCo comprennent :

1. **Couverture d'instructions** : JaCoCo peut indiquer le nombre d'instructions Java bytecode qui ont été exécutées par vos tests.
2. **Couverture de branches** : JaCoCo peut montrer la couverture des branches de vos instructions if et switch.
3. **Couverture de lignes** : JaCoCo peut indiquer quelles lignes de votre code ont été exécutées par vos tests.

# Découverte et utilisation des outils de qualimétrie dans le monde Java

# Intégration des outils de qualimétrie dans une PIC.

- L'intégration de ces outils d'analyse de code statique et de couverture de code dans un système d'intégration continue (Continuous Integration, CI) peut grandement améliorer la qualité du code, en aidant à identifier et à résoudre les problèmes tôt dans le cycle de développement.

1. SonarQube :
    - SonarQube peut être intégré à de nombreux serveurs CI, comme Jenkins, GitLab CI/CD et GitHub Actions. Vous pouvez ajouter une étape dans votre pipeline CI pour exécuter l'analyse SonarQube. Dans le cas de Jenkins, vous pouvez utiliser le plugin Jenkins SonarQube pour automatiser cette tâche.
2. Checkstyle, PMD, FindBugs,JaCoCo :
    - Ces outils peuvent être exécutés comme une partie de votre build Maven. Vous pouvez ajouter une étape dans votre pipeline CI pour exécuter mvn clean verify (ou une autre commande Maven qui exécute les plugins correspondants). Les rapports générés par ces outils peuvent ensuite être recueillis et affichés par votre serveur CI.