

R5.A8.D7 : Qualité de Développement Feuille TD-TP n° 4

Analyse de code

Objectifs :

1.- Se familiariser avec des outils d'analyse de code, propres à IntelliJ ou extérieurs, comme Sonarlint et SonarCloud

Ressources à votre disposition :

- L'archive `Tennis.zip`
- L'archive `Produit.zip`
- Le fichier `modeleTP4.doc`

Travail à faire

Exercice 1

1.- Ouvrir le projet Tennis du TDTP3

2- Analyse du code avec IntelliJ

- Analyser votre code avec l'outil d'analyse incorporé dans IntelliJ
- Comparer les problèmes identifiés par l'outil avec ceux que vous avez trouvés vous-même et que vous avez consignés dans le compte-rendu d'analyse du TD-TP3 n° 3 (point n°2)
- Dans votre compte-rendu d'analyse de TP3, ajouter un point 4.- intitulé « 4.- Noter ci-dessous les résultats de votre analyse de code par IntelliJ »
- Faire une capture d'écran de l'analyse et l'insérer dans votre rapport d'analyse de code.

3.- Créer un dossier `r5.A08.D07\tdtp4`

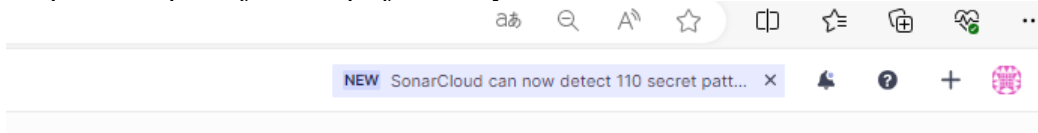
4- Analyse et correction du code avec Sonarlint

- Télécharger le projet `Tennis.zip` (cf. TP n°3) et le décompresser dans le dossier `r5.A08.D07\tdtp4`
- Renommer le projet `TennisSonarLint`.
- Créer un dépôt local dans le dossier du projet. Vérifier que le projet est ok (ouvrir IntelliJ et exécuter `main()`)
- Installer le plugin SolarLint sur l'IDE IntelliJ :
Sur IntelliJ — aller à :
File >> Settings >> Plugins >> Type 'SonarLint' >> Install and Restart IDE.
- Analyser votre code avec SonarLint
 - Lancer l'analyse du code avec SonarLint
 - Comparer les problèmes identifiés par l'outil avec ceux que vous avez trouvés vous-même et que vous avez consignés dans le compte-rendu d'analyse du TD-TP3 n° 3 (point n°2)
 - Dans votre compte-rendu d'analyse, ajouter un point 5.- intitulé « 5.- Noter ci-dessous les résultats de votre analyse de code par SonarLint »
 - Faire une capture d'écran de l'analyse de SonarLint et l'insérer dans votre rapport d'analyse de code.
- Corriger les problèmes de code détectés puis engager (commit) sur le dépôt local et le pousser sur le dépôt distant.

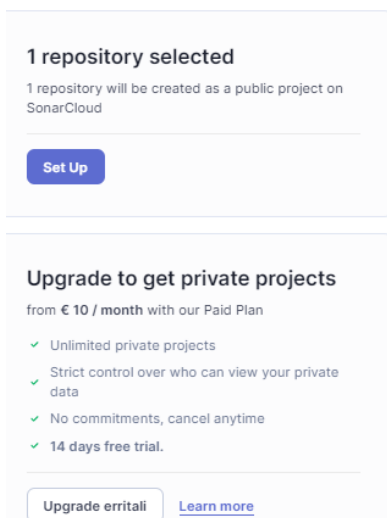
- g. Analyser le couplage (dépendances) de la classe ShoppingCart et représenter les dépendances avec un diagramme

5- Analyse avec SonarCloud

- Télécharger le projet **Tennis.zip** du TP 3 et le décompresser dans le dossier **r5.A08.D07\tdtp4**
- Renommer le projet **TennisSonarCloud**.
- Créer un dépôt local dans le dossier du projet. Vérifier que le projet est ok (ouvrir IntelliJ et exécuter main())
- Aller à l'adresse <https://sonarcloud.io/login> et utiliser votre compte github pour la création de compte sur SonarCloud et faire les configurations nécessaires pour l'import du projet
 - Cliquer sur + pour ajouter le projet à analyser sur SonarCloud



- Choisir votre repository puis cliquer sur set up et finaliser la configuration



- Analyser, corriger quelques-unes des erreurs détectées (au moins 2 par catégorie identifiée par SonarCloud), puis engager (commit) la nouvelle version

Pour avoir plus d'information sur les métriques utilisées par SonarCloud :

[https://docs.sonarsource.com/sonarqube/9.9/user-guide/metric-definitions/#:~:text=Security%20review%20rating%20\(%20security_review_rating%20\)%3A,as%20Acknowledged%2C%20Fixed%20or%20Safe](https://docs.sonarsource.com/sonarqube/9.9/user-guide/metric-definitions/#:~:text=Security%20review%20rating%20(%20security_review_rating%20)%3A,as%20Acknowledged%2C%20Fixed%20or%20Safe)

- Configurer GitHub action pour cette repository afin de générer le rapport de couverture de code (cf. révisions TD-TP n°1).

Exercice 2

1.- Installation des fichiers, création d'un dépôt pour la gestion de versions

- Télécharger l'archive **Produit.zip** disponible sur eLearn. Décompresser l'archive.
- Déposer le dossier décompressé dans **r5.A08.D07\tdtp4** puis supprimer l'archive **.zip** téléchargée.
- Lancer IntelliJ, ouvrir le projet **Produit**
- Si l'exécution est ok, engager (commit) le projet avec cette version initiale

2.- Analyse et correction du code

- a. Analyser le code manuellement puis avec Sonarlint. Pour chaque analyse, noter les problèmes détectés sur le compte-rendu d'analyse fourni
- b. Créer un dépôt sur github et analyser le même code avec SonarCloud
- c. Analyser le couplage du projet avec l'outil intégré à IntelliJ. Noter les résultats sur le compte-rendu d'analyse de code.

3.- Corriger et tester avec rapport de couverture de code

- a. Corriger les problèmes du code et définir les tests pour avoir une bonne couverture
- b. Faire un deuxième commit après l'ajout de vos tests et revenir sur SonarCloud pour analyser les résultats, que vous comparerez aux résultats de la première analyse
- c. Générer le rapport de couverture de code avec IntelliJ