#### IUT de BAYONNE PAYS BASQUE - DÉPARTEMENT INFORMATIQUE

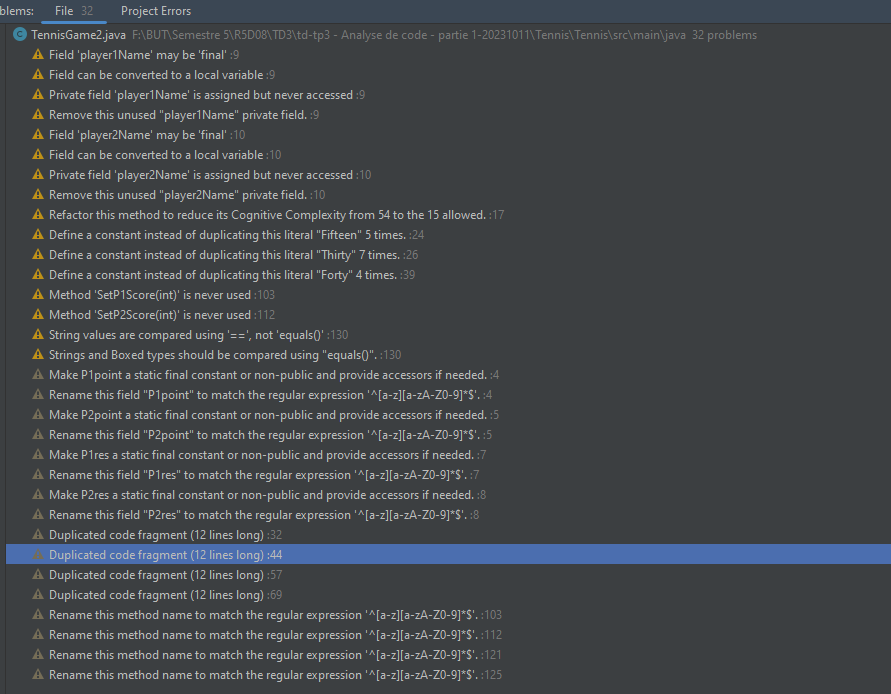
Semestre 5- 2023-2024

**R5.A8.D7 : Qualité de Développement   
TD-TP n° 3**

Analyse de code

1. Enregistrer ce document avec votre nom
2. Noter les résultats de votre analyse manuelle sur le tableau suivant :
3. Problèmes sur l’application Tennis

|  |  |  |
| --- | --- | --- |
| Problème | Ligne | Bonne pratique à utiliser |
| Règle de nommage variable | 4, 5,7,8 | camelCase (exemple : pointP1) |
| Règle de nommage fonction | 121 et 125 | camelCase(exemple : scoreP1()) |
| Code mort | 9 et 10 |  |
| Règle de taille de code | De 21 à 80 | Mettre des switch() à la place des if{} quand on cherche avec quoi une variable est égale. |
| Code mort | 92 et 96 | Mettre une vérification d’un nombre qui est minimum à 0 ne sert à rien |
| Code mort | 52 du fichier test | Mettre une fonction de vérification dans le fichier test ne sert à rien, le mettre dans le fichier de jeu serait mieux |
| Règles sur la documentation javadoc |  | Mettre une javadoc serait mieux pour appréhender le code, dans le pire des cas mettre des commentaires. |
|  |  |  |



1. Le graphe de contrôle et calculer la complexité cyclomatique

CC = E - N+p

91 – 62+2 = 31

1. La métrique de cohésion de la classe TennisGame2

|  |  |  |
| --- | --- | --- |
| attribut | Méthode() | Nombre |
| P1point | getScore()  P1Score() | 2 |
| P2point | getScore()  P2Score() | 2 |
| P1res | getScore() | 1 |
| P2res | getScore() | 1 |
| player1Name | TennisGame2() | 1 |
| player2Name | TennisGame2() | 1 |

([(2 + 2 + 2)/3 – 2]/[1 – 2])

[(a1+..+) /NbA-NbM]/[1-NbM) = cohésion

((2+2+1+1+1+1)/ 6–7 ) / (1- 7)) = 0.94444444444

1. Le facteur d’encapsulation AHF de TennisGame2