Projet 2 : Débuggez une application Java

1. Présentation de l'application (1 minute)

Contexte *: Stage chez Heme Biotech, petite entreprise du secteur pharmaceutique.*

Fonctionnalités principales *: Travail sur une application qui lit des symptômes dans un fichier, les compte et génère le résultat dans un fichier.*

Mission : *Correction/Amélioration*

2. Analyse du code existant et de ses problèmes (1-2 minutes)

Observation initiale *: Le code ne fonctionne pas pour 2 des 3 symptômes recherchés, donne des 0.*

Identification des problèmes *:*

- utilisation de mauvaise variable ou erreur d'argument

- Manque de lisibilité

- Tout est regroupé dans une seule fonction main

- Ne compte pas tous les symptômes

Maintenant que les problèmes sont identifiés, passons à la méthode

3. Méthode (1-2 minutes)

Suivre les étapes prévues

Décomposer en effectuant des tests pour chaque modification

Chercher à produire un code compréhensible et suivant les normes : nom, camelCase, commentaires…

Versioning Git : clone / pull commit push / add rm / checkout branch merge

4. Solutions et modifications du code (2 minutes)

(code1)

Correction du code *: -pourquoi pupils fonctionne et les autres non : rush/rash, head/headache*

-commentaires et déclarations inutiles

(code2)

Restructuration de la classe principale *(`AnalyticsCounter`) en plusieurs méthodes plus petites, chacune accomplissant une tâche spécifique (lecture des données, comptage de* ***TOUS*** *les symptômes, tri, écriture des résultats).*

(code2/3)

Interface pour l'écriture des données *: code plus propre, cloisonné et évolutif(changement possible dans WriteSymptoms sans impacter AnalyticsCounter)*

Javadoc

5. Tests (1 minute)

Tests pour chaque fonctionnalité
Résultats concluants *: valide le bon fonctionnement de l'appli*

6. Difficultés rencontrées (1 minute)

Pas vraiment de difficultés (vsc et les extensions s'occupent de pas mal de choses):
Au tout début suite au clone, path
Partir d'un code existant mal fait
Concept d'interface

7. Axes d'amélioration (1-2 minutes)

- **Améliorations potentielles** :
    - Proposez quelques idées pour rendre le programme plus performant ou plus complet, telles que :

Historique *-> analyse des tendances*
Interface graphique *pour rendre plus accessible et agréable*
Support de différents formats de fichiers d'entrée/sortie