

TP1 – String Toolkit – Énoncé

Objectifs

- Pratiquer **JUnit 5** (tests unitaires) et **JaCoCo** (couverture).
- Implémenter des utilitaires de chaînes robustes et testables.
- Préparer un dépôt propre (README, Maven Wrapper).

Livrables attendus

1. URL du dépôt Git (public ou accès fourni).
2. Code source des 3 services implémentés.
3. Jeux de **tests JUnit 5** couvrant les cas nominaux + bords + erreurs.
4. Capture d'écran du rapport **JaCoCo** (index.html) après `mvnw verify`.
5. README minimal : comment exécuter les tests et où trouver le rapport.

Contexte & périmètre

Vous devez coder dans `src/main/java/edu/tp1/toolkit/` :

- **Slugifier** : transformer un texte libre en *slug* (minuscules, accents retirés, ponctuation nettoyée, espaces → `-`, multi-`-` réduits, trim).
- **PalindromeService** : `isPalindrome(String)` → `true` si la chaîne est un palindrome **en ignorant** la casse, les espaces, la ponctuation et les accents.
Convention : vide/espaces → palindrome.
- **WordCounter** : `count(String)` → nombre de mots en considérant espaces, tabulations, `-` et `_` comme séparateurs ; ponctuation classique retirée ; accents considérés comme lettres. `null` → `IllegalArgumentException`.

Les signatures existent déjà ; **ne les changez pas**.

Étapes (guideline)

1. **Clone & build** : `./mvnw -B -V clean verify` (ou `mvnw.cmd` sous Windows).
2. **Créer la structure de tests** (`src/test/java/...`), ajouter **JUnit 5** (déjà dans `pom.xml`).
3. **Implémenter progressivement** chaque service :
 - commencer par les cas nominaux ;
 - gérer `null`, vide, accents, ponctuation, séparateurs multiples ;
 - préférer des méthodes courtes et testables.
4. **Écrire des tests** : cas positifs/négatifs, bords (`null`, vide, multi-séparateurs, idempotence...).
5. **Mesurer & améliorer la couverture** avec **JaCoCo** : ouvrir `target/site/jacoco/index.html` et combler les branches non couvertes.
6. (**Option Jenkins**) : si disponible, utilisez le `Jenkinsfile` fourni pour exécuter `mvnw verify` en CI.

Règles

- **Ne pas** modifier les versions Maven/JUnit/JaCoCo.
- **Ne pas** supprimer le Maven Wrapper.
- Code clair, testé, **sans warnings** de compilation.

Bon TP !