# Activité 1 :

Suivre les étapes suivantes et enregistrer le résultat affiché dans la console Git après chaque étape :

1. Créer un dossier sur votre ordinateur nommé « M14\_AP05\_01 »
2. Copier le projet Java à l’intérieur du dossier ;
3. Initialiser le dépôt Git et commit vers le dépôt local ; (message 1)
4. Créer une nouvelle classe **Cylindre.java** avec les autres modèles ;
5. Ajouter le fichier au Staging Area (index) ;
6. Afficher les différences entre index-repository
7. Retirer le fichier du staging Area
8. Afficher les différences entre index-repository
9. Ajouter la ligne **classe Cylindre** au fichier ;
10. Ajouter le fichier au dépôt local (message 2)
11. Créer une nouvelle classe **Pyramide.java** avec les autres modèles ;
12. Ajouter le fichier au dépôt local avec le commit précédent et non pas un nouveau commit ;
13. Afficher la liste des commit et vérifier qu’il n’y a que 2 commit et non 3 ;
14. Ajouter la ligne **private int rayon** au fichier **Cylindre.java** et enregistrer-le ;
15. Rétablir la version du Staging Area. *Ouvrir le fichier pour vérifier* ;
16. Ajouter la ligne **private int rayon** au fichier **Cylindre.java** ;
17. Publier les modifications vers le dépôt local ; (message 3)
18. Ajouter la ligne **private int hauteur** au fichier **Cylindre.java** ;
19. Publier les modifications vers le dépôt local ; (message 4)
20. Afficher les commits effectués avec la commande : **git log --pretty=format:"%h %s" --graph**
21. Créer un nouveau commit qui annule les commits jusqu’au **SHA1** du **message 2** tout en gardant l’historique.

### Rappel :

**git revert SHA1**

**git revert HEAD** *permet de faire un revert sur la version précédente*

Il va vous afficher une erreur car avant ce commit il n’y avait pas les fichiers **Cylindre.java** et **Pyramide.java.**

Vous avez deux possibilités :

1. Annuler le revert avec **git revert --abort**
2. Régler le conflit en retirant les fichiers puis valider le revert :

**git rm src/dz/etm/formation/model/Cylindre.java**

**git rm src/dz/etm/formation/model/Pyramide.java**

**git revert --continue**

1. Dérouler la première possibilité. Puis refaire l’étape 20 et dérouler la deuxième possibilité. Vérifier que les deux fichiers n’existent plus.
2. Afficher les commits avec la commande utilisée plus haut. S’assurer qu’il existe 5 commit ;
3. Faire un **revert** vers le dernier commit

# Activité 2 :

Suivre les étapes suivantes et enregistrer le résultat affiché dans la console Git après chaque étape :

1. Créer un dossier sur votre ordinateur nommé « M14\_AP05\_02 »
2. Copier le projet Java à l’intérieur du dossier ;
3. Initialiser le dépôt Git et commit vers le dépôt local ; (message 1)
4. Créer une nouvelle classe **Pentagone.java** avec les autres modèles ;
5. Ajouter le fichier au dépôt local (message 2)
6. Créer une nouvelle classe **Hexagone.java** avec les autres modèles ;
7. Ajouter le fichier au dépôt local (message 3)
8. Créer une nouvelle classe **Heptagone.java** avec les autres modèles ;
9. Ajouter le fichier au dépôt local (message 4)
10. Afficher les commits avec la commande : **git log --pretty=format:"%h %s" –graph**. S’assurer qu’il y a 4 commits
11. Créer deux autres copies du dossier. Nommer-les « M14\_AP05\_03 » et « M14\_AP05\_04 »

### Pour le dossier M14\_AP05\_02 :

1. Changer le HEAD vers le commit (message 2) en écrasant le contenu du dépôt local seulement.
2. Vérifier la différence entre le staging area et le dépôt local
3. Afficher les commits

### Pour le dossier M14\_AP05\_03 :

1. Changer le HEAD vers le commit (message 2) en écrasant le contenu du dépôt local et l’index.
2. Afficher les commits
3. Afficher l’état du dépôt git

### Pour le dossier M14\_AP05\_04 :

1. Changer le HEAD vers le commit (message 2) en écrasant le contenu du workspace
2. Afficher les commits
3. Vérifier le contenu du workspace