

Modifica di un progetto Java

In questa esercitazione sei uno sviluppatore che deve effettuare una nuova modifica al progetto aziendale sviluppato in Java, in particolare deve aggiungere una nuova funzionalità a una classe. Devi come prima cosa fare una fork del progetto, scaricartelo in locale, fare le modifiche, farne la commit, la push e infine la pull request. Se le tue modifiche vengono approvate dal capoprogetto, verranno aggiunte al branch principale (*main*).

Più in dettaglio, i passi da effettuare sono:

1. Creare una *fork* del progetto <https://github.com/carmelo-cina-sal/Java> su GitHub. Se l'operazione è stata fatta correttamente, dovrai trovarti il progetto sul tuo repository.
2. Effettuare la *git clone* della nuova repository sul tuo PC locale, sotto *c:\Utenti\studente\PortableGit\repository*.
3. Implementare uno dei metodi di una delle 4 classi Java presenti. Le regole per la modifica sono le seguenti:
 - a. Chi si trova nella prima fila, modificherà la classe *Fila1.java*; chi si trova nella seconda fila modifica *File2.java* ecc.
 - b. Chi si trova più a destra (a sinistra dalla cattedra) implementerà la prima funzionalità (somma tra numeri interi), il secondo a seguire implementerà la seconda funzionalità (differenza) e così via fino alla sesta (cubo di un intero).
4. Per poter aggiungere tali modifiche, sarà necessario utilizzare il comando *git add*.
5. Al fine di allineare il tuo repository remoto, effettua la *commit* sul tuo repository locale e infine effettua la *push* in remoto.
6. Su GitHub, dalla copia del tuo repository, effettua la Pull request al progetto principale (ricorda di inserire una descrizione che spieghi quale funzionalità hai aggiunto).

Di seguito uno schema di riepilogo dei comandi principali di Git.

Configurazione globale Configurazione dell'utente valida per tutti i repository <code>\$ git config --global user.name "[name]"</code> Imposta il nome che vuoi mostrare sulle tue commit <code>\$ git config --global user.email "[email address]"</code> Imposta l'email che vuoi mostrare sulle tue commit	Creare repository Crea un nuovo repository o clonane uno esistente da un URL <code>\$ git init [project-name]</code> Crea un nuovo repository locale con il nome specificato <code>\$ git clone [url]</code> Scarica un progetto esistente e il suo storico di cambiamenti
Effettuare modifiche Rivedi i cambiamenti al codice e prepara una commit <code>\$ git status</code> Elenca tutti i file nuovi o modificati <code>\$ git diff</code> Mostra le differenze non ancora nell'area di staging <code>\$ git add [file]</code> Crea uno snapshot del file in preparazione al versioning <code>\$ git diff --staged</code> Mostra le differenze tra staging e ultima modifica <code>\$ git reset [file]</code> Rimuovi un file dall'area di staging, ma mantieni le modifiche	Rivedere lo storico Esplora l'evoluzione dei file del progetto <code>\$ git log</code> Elenca lo storico di versione per il branch corrente <code>\$ git log --follow [file]</code> Elenca lo storico di versione per il file specificato, incluse rinominazioni <code>\$ git diff [first-branch]...[second-branch]</code> Mostra la differenza tra due branch <code>\$ git show [commit]</code> Mostra i metadati e i cambiamenti della commit specificata

<pre>\$ git commit -m "[descriptive message]"</pre> <p>Salva gli snapshot dei file in maniera permanente nello storico</p>	
<p>Annullare commit Elimina errori e altera lo storico dei cambiamenti</p> <pre>\$ git reset [commit]</pre> <p>Annulla tutte le commit effettuate dopo [commit], preservando i cambiamenti locali</p> <pre>\$ git reset --hard [commit]</pre> <p>Elimina tutto lo storico e i cambiamenti fino alla commit specificata</p>	<p>Sincronizzare i cambiamenti Collegati a un URL remoto e ottieni lo storico dei cambiamenti</p> <pre>\$ git fetch [remote]</pre> <p>Scarica lo storico dei cambiamenti dal repository remoto</p> <pre>\$ git merge [remote]/[branch]</pre> <p>Unisci il branch remoto con quello locale</p> <pre>\$ git push [remote] [branch]</pre> <p>Carica tutti i cambiamenti dal branch locale su GitHub</p> <pre>\$ git pull</pre> <p>Scarica lo storico e unisci i cambiamenti</p>