ECUE6501

Techniques de programmation avancée

**TP Git & Design Pattern**

Git Fork : création d’une copie d’un dépôt, indépendant du dépôt initial.

Exemple / exercice

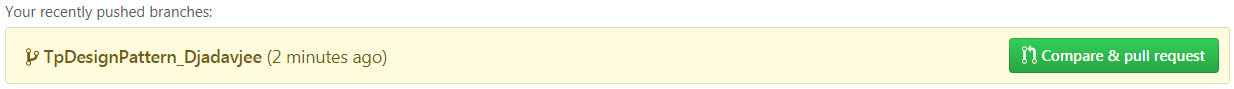
1. Se connecter sur son GitHub
2. Créer un fork du dépôt suivant : <https://github.com/DorianDdj/ECUE6501_TP_DesignPattern>
   1. Cliquer sur le bouton “Fork” pour créer une copie sur son propre dépôt



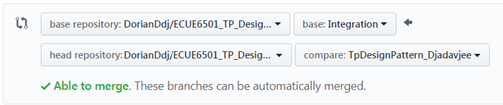
* 1. Cloner le nouveau projet
  2. Créer une branche nommée “TpDesignPattern\_VotreNom” (en remplaçant “VotreNom” par … votre nom)

1. Faire l’exercice décrit dans le fichier “ProjectDescription.txt”
2. Faire un commit puis un push sur son dépôt
3. Faire une demande “Pull Request” de son dépôt vers la branche “Integration” du dépôt initial
   1. Dans l’interface GitHub de votre dépôt, cliquer sur “New Pull Request” (ou “Compare & pull request”)

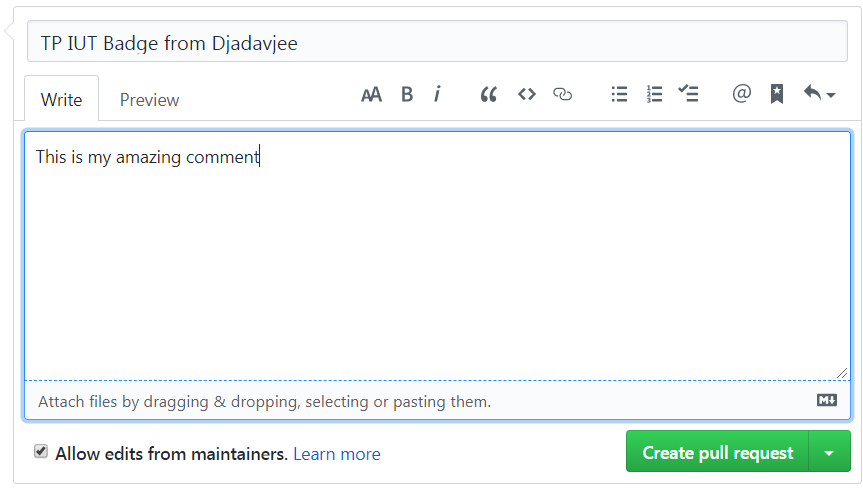




* 1. Sélectionner le dépôt initial en base, avec la branche “Integration”
  2. Sélectionner votre dépôt en head repo, avec votre nouvelle branche



* 1. Entrer un titre et une description
  2. Valider en cliquant sur “Create pull request”



Exercice 2 : Design Pattern “Observer” en Java

Le design pattern “Observer” est disponible nativement en Java :

java.util.Observable : le sujet observé, qui notifie les observers

java.util.Observer : l’observer, qui est notifié

Exercice : modifier l’implémentation de l’exercice précédent en utilisant l’Observer java

Indications :

* le sujet hérite de “Observable”, et contient une méthode qui fait appel à “notifyObservers”
* les observers implémentent l’interface “Observer” et contiennent une méthode “update(Observable observable, Object arg)”
* L’ajout des observers au sujet se fait par la fonction “addObserver”
* La notification des observers ne fonctionne que si l’on a indiqué que le sujet avait été modifié, en utilisant “setChanged();”

Extra : self improvement!

<https://www.codingame.com/>