

Documents autorisés : aucun - Durée 2h.

AVERTISSEMENT : pour les 2 problèmes, s'en tenir exclusivement au texte, sans tenir compte de ce que vous connaissez des systèmes évoqués.

PROBLÈME 1 Gestionnaire de versions

1) Diagramme états-transitions : une branche d'un dépôt local Git (5 pts)

Git est un gestionnaire de versions décentralisé (DVCS). Il permet de conserver l'historique des modifications de plusieurs fichiers d'un projet regroupés en dépôt. Il existe un dépôt distant pour le projet, et chaque membre possède une copie locale de ce dépôt. Git peut être manipulé à travers des commandes simples. Dans ce contexte on appelle branche une évolution particulière du projet. On étudie ici les **états successifs d'une branche d'un dépôt local**.

Git permet à chaque membre du projet de travailler et de sauvegarder l'historique des modifications qu'il opère au sein de son dépôt local, indépendamment du dépôt distant. On considère que le dépôt local part d'un état « clean » : les fichiers n'ont pas été modifiés. Après que le développeur a effectué une (des) modification(s), le dépôt local n'est plus à jour (divergence). Pour sauvegarder une modification, il suffit d'exécuter une commande «commit» en l'accompagnant d'un message descriptif. Quand des changements ont été sauvegardés par « commit », on dit que le dépôt local est « en avance » sur le dépôt distant. On peut partager les « commits » vers le dépôt distant grâce à la commande « push ». Lorsque tous les changements sont sauvegardés en local, le dépôt local est dans l'état « clean ».

Lorsque le dépôt local n'est pas en cours d'exécution d'un « push », on peut vérifier si des modifications ont été opérées par d'autres membres du projet en appelant la commande « pull ». Si des changements sont constatés, le dépôt local est considéré non à jour et l'on récupère les modifications. Dans tous les cas, après un pull, le dépôt local revient dans l'état précédent la tentative de mise à jour.

2) Diagramme de séquences système : publication des modifications d'un projet Git (5 pts)

Pour ajouter une fonctionnalité à un projet géré par Git, les développeurs ont pour habitude de créer une branche pour chaque fonctionnalité, de la mettre au point, puis de fusionner leur branche avec la branche principale.

Représenter à l'aide d'un diagramme de séquences **détaillé** le processus suivant :

- Le développeur crée une nouvelle branche sur le dépôt local pour sa fonctionnalité.
- Il se place dans cette nouvelle branche grâce à la commande « checkout ».
- Une fois les fichiers modifiés (*faire abstraction de cette étape sur le diagramme*) le développeur réalise un « commit » sur sa branche **pour chaque fichier modifié**. Les « commits » sont accompagnés d'une description.
- Avant de pouvoir exécuter un « push », le développeur récupère dans son dépôt local les derniers changements sur le dépôt distant par la commande « fetch ».
- Il fusionne alors sa branche avec la branche principale de son dépôt local (*considérer qu'aucun conflit n'apparaît*).
- Il reste au développeur à synchroniser sa branche principale avec celle du dépôt distant à l'aide de la commande « push » (*une fois de plus, faire abstraction d'éventuelles modifications entre la fusion et l'envoi des modifications vers le dépôt distant*).

Placer sur le diag. : le développeur (ou une IHM) et une instance de Dépôt Local, Branche, Commit et Dépôt Distant.

PROBLÈME 2 Classes (8pts) et Cas d'Utilisation (2pts) : gestion des jeux olympiques d'hiver

Les jeux olympiques sont organisés en épreuves qui appartiennent à différentes disciplines. Chaque épreuve appartient à une discipline (ski, patinage, snowboard...) et une seule.

Chaque épreuve a lieu dans une station de la région. Un même station peut accueillir plusieurs épreuves, mais à des jours différents. On connaît le nom et l'altitude de chaque station. C'est un responsable général qui décide dans quelle station aura lieu une épreuve donnée.

Les personnes responsables des jeux sont organisées de façon hiérarchique. Au sommet se trouve le responsable général, dont dépendent les responsables de disciplines. Un responsable s'occupe de plusieurs épreuves dans la même discipline, plusieurs responsables sont affectés à l'organisation d'une épreuve. C'est le responsable général qui affecte les responsables à chaque épreuve, un responsable d'épreuve affecte les numéros de dossards et les horaires des épreuves.

Chaque concurrent représente un pays et participe à une ou plusieurs épreuves. On distingue les concurrents hommes et femmes car les épreuves ne sont pas mixtes. Les concurrents sont identifiés grâce à un numéro d'inscription aux jeux et ont un nom, une date de naissance et des coordonnées. Le sélectionneur de chaque pays enregistre les concurrents.

Un concurrent peut pratiquer plusieurs disciplines, il porte un numéro de dossard différent pour chaque épreuve. Les responsables d'une épreuve enregistrent les résultats des concurrents : le classement, les médailles et, selon l'épreuve, le nombre de points (p ex. patinage) ou le temps (p. ex. ski).

Les concurrents peuvent utiliser l'application en consultation (par exemple pour connaître le planning des épreuves) et peuvent aussi modifier leur profil. Étant donné que différents niveaux d'autorisation sont nécessaires pour les différents utilisateurs, chacun devra s'identifier pour utiliser l'application.