RAPPORT DE PROJET

Projet design pattern

Membre du groupe

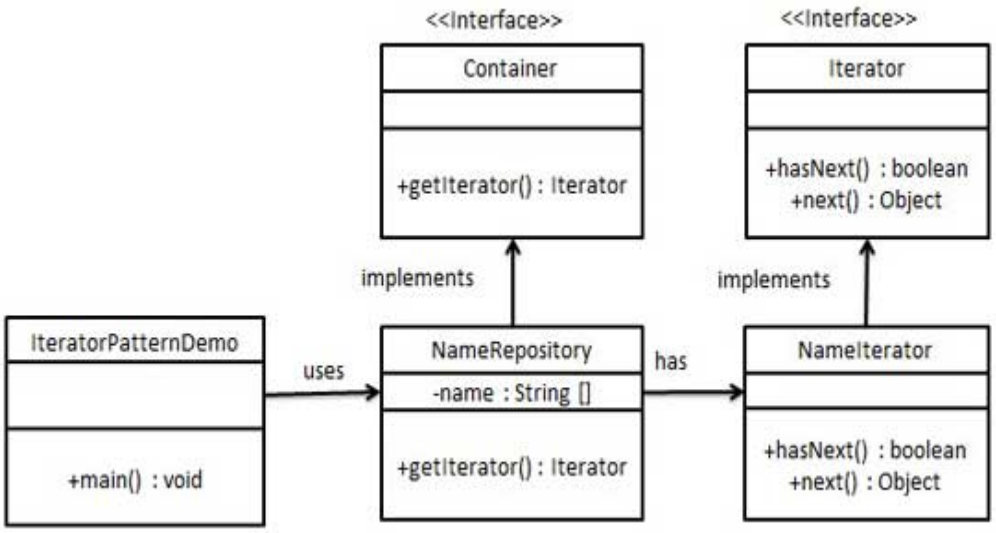
SIDIBE Kadiatou
GUEYE Bassirou
YAHYA BEY Sami
VANIE Jean-Marc
ALLEGOU Bachi
TRAORE Abdoul-Karym

Objectif: réorganiser un le code d'un outil permettant de récupérer des donnés text afin d'obtenir une vision claire des données.

Le projet est constitué de 7 questions à travers lesquelles on effectue des opérations, vérifications, réorganisations etc. La réponse à ces questions nécessite ou pas l'utilisation d'un design pattern. Ce rapport s'accentue sur les questions dont la réponse nécessite un design pattern.
Pour chaque design pattern utiliser une explication du choix est donné ainsi d'un diagramme de classe illustrant le fonctionnement du design pattern.

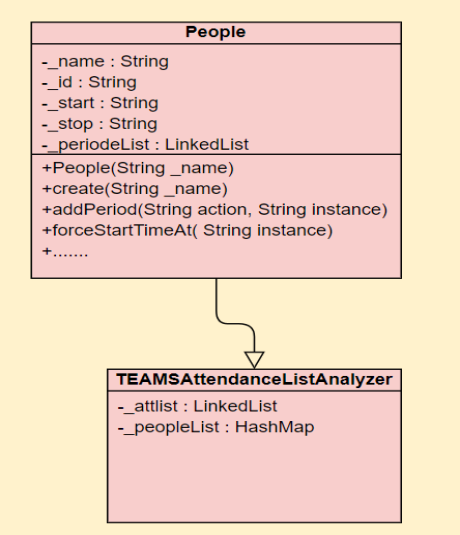
Iterator

Dans la question numéro 2 nous avons utilisé le Design pattern Iterator pour modifier les parcours de collection de liste. Nous avons choisi ce design pattern car Itérateur est un patron de conception comportemental qui permet de parcourir les éléments d'une collection sans révéler sa représentation interne (liste, pile, arbre, etc.). Les "iterator" on été utilisés dans la classe TEAMSProcessor sur la méthode (String toHTMLCode), dans la classe TEAMSAttendanceListAnalyzer sur la méthode (void setStartAndStop), dans la classe People sur la méthode (getTotalAttendanceDuration) et sur la méthode (String getHTMLCode).

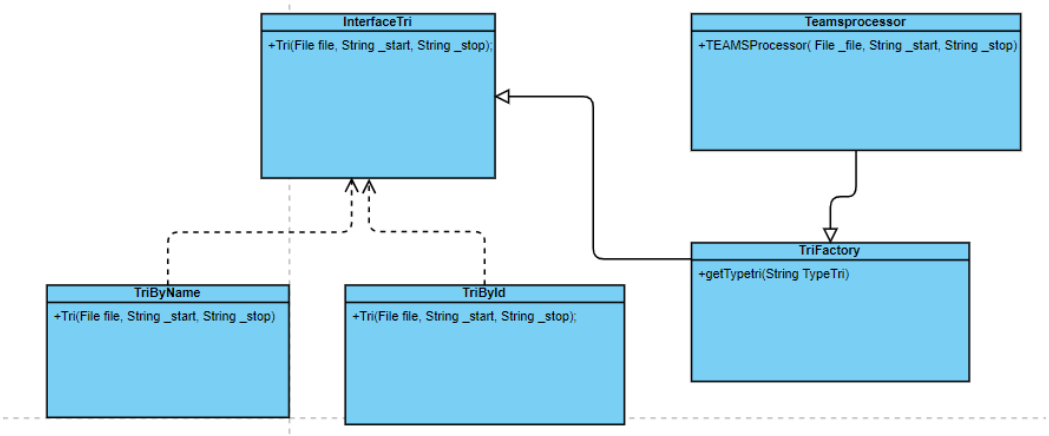


Factory

Pour la réponse à la question 4 comme demandé nous avons utilisé le design pattern Factory pour instancier les objets de la classe “People”. Pour ce faire nous avons créé une méthode factory à travers laquelle on crée un objet “People” ensuite qui peut être instanciée dans différentes classes sans faire appel au constructeur. Puis nous avons passé la visibilité du constructeur à private.



Nous avons également utilisé les factory pour une autre tâche de du projet. Pour effectuer le trie nous avons eu besoin du design pattern factory pour instancier les classes “TrieById” et “TrieParNom” implémentant l’interface trie (via la classe "TriFactory''). Ce design est représenté comme suit:



Stratégie

Quant à la question 5, l'inclusion du MVC nous a amené à repenser les tri par id , par nom ou par durée de connexion. Pour ce faire nous avons implémenté le design pattern Strategy de sorte que, selon le bouton radio bouton(id,nom) cliqué le traitement correspondant s'applique à la sortie en HTML.