

# Gestion produits

On se propose de développer un programme qui permet de gerer les prestations d'une entreprise.

Une **prestation** identifié par son id (int), nom\_projet (String) et nom\_client (String), peut etre soit un **produit** caracterisé par prix\_total (int) et date\_livraison (String), ou bien un service caracterisé par nombre\_heures (int) et prix\_heure (int)

Pour se faire, vous devez suivre les instructions en complétant les TODOs en prenant en compte qu'une **entreprise** contient plusieurs prestations (List<Prestation>)

**TO DO 1 :** Créer les classes entreprise, prestation, produit et service en respectant la notion de l'heritage et la classe scellés sachant que produit et service ne sont pas heritables

**TO DO 2 :** Pour chaque classe, implementer un constructeur vide, un constructeur paramétré, les getters et les setters, ainsi que la redefinition de toString, hashCode et Equals sachant 2 objects considéré egaux s'ils ont le meme id.

**TO DO 3 :** Créer la méthode « ajouterPrestation(Prestation p) » qui permet d'ajouter une prestation à l'entreprise sachant que si une prestation existe deja, une Exception de type PrestationExisteException sera levée.

**TO DO 4 :** implémenter la méthode « void afficherNbrPrestation() » qui permet d'afficher le nombre de prestation pour chaque type

**TO DO 5 :** implémenter la méthode « List<Prestation> prestationparClient(String client) » qui permet de renvoyer la liste des prestations concernant le client donné, une liste vide sinon.

**TO DO 6 :** implémenter la méthode « Map<String, List<Prestation>> filterParClient() » qui permet de renvoyer la liste des nom\_projet par rapport aux clients.

**TO DO 7 :** Renvoyer la liste des clients d'une entreprise avec la methode «`List<String> afficherClient()`», et renvoyer la liste des noms de projets d'une entreprise avec la methode «`List<String> afficherProjects()`». Assurer vous que les clients et les projets soit triés et non-redondants.

**TO DO 8 :** Créer la methode «`Map<Service, Integer> recupererServices()`» qui retourne les services et leurs prix totaux, et qui utilise un object anonyme de `Function<Service, Integer>` qui permettra de calculer le prix total d'un service [`prix_heure * Nombre_heure`].

**TO DO 9 :** Créer la methode «`void totalVentesProduits()`» qui affiche le total des ventes des produits de l'annee 2023

Hint : Stream

**TO DO 10 :** Créer la methode «`void topServiceClient()`» qui renvoi le meilleur client, celui qui a généré le maximum de benefice.

Hint : Stream