 2024 -2025

**TP 1 - JAVA**

**Exercice 1: Classe, constructeurs, méthodes et attributs**

1. Créer un nouveau projet de type *Java application* (package : Ex1et classe : Personne)
2. Créer une classe **Personne** avec les attributs (private? protected?...) :

* **nom** (chaîne de caractères),
* **prenom** (chaîne de caractères);
* **anNaissance** (entier)

1. Dans la classe Personne créer:

* un constructeur naturel permettant d'initialiser le nom, le prénom et la date de naissance avec les paramètres fournis
* un constructeur permettant d’initialiser nom à "Inconnu" et qui récupère prenom et année de naissance dans les paramètres du constructeur
* un constructeur par défaut (appeler le premier constructeur) où :
  + nom = "Potter"
  + prenom = "Harry"
  + anNaissance = 1980

1. Définir les 3 getters : getNom,…. (pour les 3 attributs)
2. Définir les 3 setters : setNom, … (pour les 3 attributs)

**Exercice 2: Suite :**

1. Définir une méthode calculerAge() permettant de retourner l'âge
2. Définir une méthode afficherInfos() qui affiche sur la console les informations d’une personne

Nom: Dupont

Prénom: Jacques

Âge: 35

1. Définir une méthode mange() qui prend en paramètre une chaine de caractères et qui affiche un message comme dans cet exemple :

Dupont Jacques mange un/une Sandwich

1. Dans une autre méthode, placez votre point d'entrée ("**main**").

public static void main(String[] args) {

// Création d'une instance de la classe Personne

Personne personne1 = new Personne("Dupont", "Jean",2004);

}

1. Créez 3 instances de la classe "Personne" en utilisant les différents constructeurs.
2. Invoquer les méthodes get(), set(),mange(), calculerAge() et afficherInfos() sur les différentes instances créées.
3. Ecrire une méthode **afficherNbPers**() qui compte le nombre d'objets déjà créés. A chaque création d'un objet elle doit nous afficher le nouveau nombre de Personnes.

**Exercice 3: Pour aller plus loin :**

1. Une année est bissextile si cette année est multiple de 4 mais pas de 100 ou si c'est un multiple de 400

Ecrire une méthode isBissextile qui prend en paramètre l’année de naissance d’une personne et retourne un booléen (true si bissextile, false sinon).

1. Ecrire une méthode qui prend comme paramètre l’année de naissance d’une personne et qui affiche tous les nombres premiers inférieurs ou égale à cette année.
2. Un palindrome (<https://fr.wikipedia.org/wiki/Palindrome>) est un mot ou une phrase qui se lit indifféremment de droite à gauche et de gauche à droite.

Ecrire une méthode isPalindrome()qui prend en paramètre une chaine de caractères et qui retourne un booléen.

Tester pour une personne si son nom, prénom (anna) ou année (2002) est un palindrome

**Exercice 4 : Gestion de Magasin en Java**

Créer un programme Java qui simule un magasin en ligne.

**Organisation des fichiers dans un package :**

* Créez un package nommé gestionMagasin.
* Dans ce package, créez les classes suivantes :
  + Produit.java
  + Client.java
  + Panier.java
  + Commande.java
  + Magasin.java
  + Main.java

**Classes à créer :**

1. **Produit** (classe représentant un produit)
   * Attributs : id, nom, prix, quantité
   * Méthodes :
     + Constructeur pour initialiser les attributs.
     + Getters et setters pour accéder et modifier les attributs.
     + afficherDetails() pour afficher les détails du produit.
2. **Client** (classe représentant un client)
   * Attributs : id, nom, email
   * Méthodes :
     + Constructeur pour initialiser les attributs.
     + Getters et setters pour accéder et modifier les attributs.
     + afficherDetails() pour afficher les informations du client.
3. **Panier** (classe représentant le panier d'un client)
   * Attributs : produits (liste des produits ajoutés au panier)
   * Méthodes :
     + ajouterProduit(Produit produit) pour ajouter un produit au panier.
     + supprimerProduit(Produit produit) pour supprimer un produit du panier.
     + afficherPanier() pour afficher les produits dans le panier.
     + calculerTotal() pour calculer le montant total du panier.
4. **Commande** (classe représentant une commande passée par un client)
   * Attributs : idCommande, client, produitsCommandes, total
   * Méthodes :
     + Constructeur pour initialiser une commande avec un client et les produits du panier.
     + afficherDetailsCommande() pour afficher les informations sur la commande.
5. **Magasin** (classe représentant le magasin)
   * Attributs : produits (liste des produits disponibles dans le magasin)
   * Méthodes :
     + ajouterProduit(Produit produit) pour ajouter un produit au magasin.
     + afficherProduitsDisponibles() pour afficher les produits disponibles.
     + trouverProduitParNom(String nom) pour trouver un produit par son nom.
6. **Main** (classe principale contenant la méthode main())
   * Méthodes :
     + Création des clients, des produits et des magasins.
     + Interaction avec le client pour afficher les produits, ajouter au panier et passer une commande.

Exemple de menu :

--- Menu Magasin ---

1. Afficher les produits disponibles

2. Ajouter un produit au panier

3. Afficher le panier

4. Passer la commande

5. Quitter