

TP Révision

Exercice 1

La classe Candidat est une classe qui modélise les candidats à une expérience scientifique. Cette classe doit permettre de calculer la masse corporelle des candidats et déterminer si le candidat en question est Mince, Normal ou Obèse.

Chaque candidat est caractérisé par :

- Son Nom et son Prénom de type chaîne de caractères.
- Son Poids et sa Taille de type double.
- Le constructeur de la classe candidat permet d'initialiser les attributs par les valeurs des variables passées en paramètre.
- La classe Candidat possède également une méthode calculMasseCorporelle() qui permet de calculer la masse corporelle d'un candidat selon la formule suivante : $\text{poids}/(\text{taille}*\text{taille})$
- Une autre méthode EstObèse() permet de renvoyer une chaîne de caractère qui indique si le candidat est Mince, Normal ou Obèse. Une personne est normale si sa masse corporelle varie entre 16 et 25. Au-dessous de 16, la personne est mince et au-dessus de 25, elle est obèse.
- La méthode Affiche() permet d'afficher les informations du candidat comme suit :

Le sujet Mekni pèse 55.0KG et mesure 1.7m

Sa masse corporelle est égale à 19.031141868512112

Le sujet est Normal

Ecrire la classe Test_Candidat pour tester la classe Candidat et afficher les 2 candidats suivants :

Le sujet Mekni pèse 55.0KG et mesure 1.7m

Sa masse corporelle est égal à 19.031141868512112

Le sujet est Normal

Le sujet Lazhar pèse 95.0KG et mesure 1.75m

Sa masse corporelle est égal à 31.020408163265305

Le sujet est Obèse

Exercice 2

Nous allons définir une classe Pays permettant de stocker et manipuler des informations sur les pays. Chaque instance de cette classe contiendra les informations suivantes concernant un pays :

- son nom,
- sa superficie,
- sa population.

Cette classe possède les méthodes suivantes :

- Un constructeur sans paramètre qui initialise les attributs : nom, superficie et population avec les valeurs null, 0 et 0 respectivement
- Un constructeur dont la signature Pays(String n, int s, int p)
- Les accesseurs : getNom(), getSuperficie() et getPopulation()
- Les modificateurs : setNom(String n) , setSuperficie(int s) la superficie doit être supérieur à 20000 km2 et setPopulation(int p)
- Une méthode toString() qui ne reçoit aucun paramètre et renvoie la chaîne de caractères de la forme : Nom : Tunisie, Superficie : 75417 km2, Population : 11 millions
- Une méthode densite() qui ne reçoit aucun paramètre et renvoie la densité du pays, en nombre d'habitants par Km², définie par la formule : $\text{densité} = (\text{population} * 1000000) / \text{superficie}$.
- Une méthode plusPeupleQue(Pays p). Cette fonction reçoit en paramètre une instance p de Pays. Elle renvoie un booléen qui vaut true si le pays auquel elle est appliquée est plus peuplé que le pays p et false sinon.
- Une méthode plusDenseQue(Pays p). Cette fonction reçoit en Paramètre une instance de Pays. Elle renvoie un booléen qui vaut true si le pays auquel elle est appliquée est plus dense que le pays p et false sinon.

Définir la classe pays avec ces attributs et ces méthodes

Modifier la classe pays pour qu'on puisse afficher le nombre de pays créés et le nombre total des populations de tous les pays créés (ajouter les attributs nécessaires et modifier les méthodes concernés)

Exercice 3

Le directeur d'une entreprise de produits chimiques souhaite gérer les salaires et les primes de ses employés vendeurs au moyen d'un programme Java.

1. Codez une classe Employé caractérisée par les attributs : nom (String), prénom (String), année d'embauche (int), salaire de base (double) et un chiffre d'affaire (double). Ces attributs ne doivent être visibles qu'à l'intérieur de la classe employé.

Et les méthodes suivantes :

- Un constructeur prenant en paramètre l'ensemble des attributs.
- Une méthode public String toString() retournant une chaîne décrivant l'employé.
- Une méthode void Affiche() qui affiche la description de l'employé à l'écran.
- Un accesseur et un modificateur pour l'attribut salaire de base.
- Une méthode boolean AccorderPrime() qui retourne true si une prime est accordée à l'employé et false sinon. Les vendeurs reçoivent une prime au cas où le chiffre d'affaire dépasse les 5000dt.
- double doubleCalculPrime() qui permet de calculer et de retourner la valeur de la prime mensuelle que l'employé va recevoir. La prime vaut 2% du chiffre d'affaire.
- Une méthode double CalculSalaire() qui calcule le salaire selon que l'employé ait reçu une prime ou pas. Cette méthode retourne le salaire de base si la prime n'est pas accordée, sinon elle retournera le salaire de base auquel on ajoute la prime.

2. Codez une classe principale test dans laquelle :

On instancie un employé. On l'affiche à l'écran. On calcule et on affiche son salaire