SERIE EXERCICES POO SOUS C#

Exercice 1 : Gestion Compte Bancaire

- Un compte bancaire possède à tout moment une donnée : son solde. Ce solde peut être positif (compte créditeur) ou négatif (compte débiteur).
- Chaque compte est caractérisé par un code incrémenté automatiquement.
- le code et le solde d'un compte sont accessibles en lecture seulement.
- A sa création, un compte bancaire a un solde nul et un code incrémenté.
- Il est aussi possible de créer un compte en précisant son solde initial.
- Utiliser son compte (épargne ou payant) consiste à pouvoir y faire des dépôts et des retraits. : méthode Deposer() et Retirer() qui demandent à l'utilisateur d'entrer le montant à retirer ou à déposer mais pour le compte payant ces opérations sont payantes est vaut 5DH (à chaque fois ou l'utilisateur retire ou dépose de l'argent de son compte payant son solde sera diminué par 5DH)
- L'utilisateur peut aussi consulter le solde de son compte par la méthode ToString().
- Un compte Epargne est un compte bancaire qui possède en plus un champ « Taux Intérêt=6 » et une méthode calculIntérêt() qui permet de mettre à jour le solde en tenant compte des intérêts.

  Solde = solde+ solde * (TauxInteret/100)

  Questions :

1. Définir la classe Compte (avec les propriétés / les constructeurs (par défaut , d'initialisation , par copie)/les méthodes)
2. Définir la classe CompteEpargne. (avec les propriétés / les constructeurs (par défaut , d'initialisation , par copie)/les méthodes)
3. Définir la classe ComptePayant. (avec les propriétés / les constructeurs (par défaut , d'initialisation , par copie)/les méthodes)
4. Créer un programme permettant de tester ces classes avec les actions suivantes:

- Créer une instance de la classe Compte , une autre de la classe CompteEpargne et une instance de la classe ComptePayant

- Faire appel à la méthode déposer() de chaque instance pour déposer une somme quelconque dans ces comptes.
- Faire appel à la méthode retirer() de chaque instance pour retirer une somme quelconque de ces comptes.
- Faire appel à la méthode calculInterêt() du compte Epargne.
- Afficher le solde des 3 comptes.

Exercice 2 : Gestion Produits informatiques :

On souhaite développer une application orienté objet pour une société de fabrication des produits informatiques. L'application contiendra plusieurs classes dont tous les attributs doivent être privés, ce qui implique le besoin de coder des accesseurs dans toutes les classes. Chaque classe doit avoir un constructeur d'initialisation et une méthode ToString() .

1 – Créer les classes suivantes : (avec les constructeurs / propriétés / méthode ToString())

- Classe **Zone :** ayant comme attribut : idZone : entier / NomZone : chaine
- Classe **Pays :** ayant comme attribut : idPays : entier / NomPays : chaine / zone : objet de ma classe Zone
- Classe **Ville :** ayant comme attribut : idVille : entier / NomVille : chaine /CodePostal : entier / pays : objet de ma classe Pays
- Classe **Catégorie :** ayant comme attribut : idCatégorie : entier / NomCatégorie : chaine / catégorie : objet de la même classe catégorie. Cette classe a un 2 ème constructeur avec un seul paramètre (nom de la catégorie).
- Classe **Produit :** ayant comme attribut : idProduit : entier / NomProduit : chaine /PrixProduit : réel/ catégorie : objet de la classe Catégorie.

2- Dans la classe Pays , ajouter un nouvel attribut **« Produits_disponibles** » qui sera une liste des produits disponible dans le pays

3- Dans la classe Pays , ajouter une méthode **« AjouterProduit(produit) »** permettant d'ajouter une nouveau produit à la liste des produits disponibles dans le pays , si on ajoute un produit qui a le meme identifiant qu'un produit déjà à la liste affichez le message « Produit existe déjà »

4- Dans la classe Pays , ajouter une méthode **« SupprimerProduit(Idproduit) »** permettant de supprimer un produit de la liste des produits disponibles par son identifiant si le produit n'existe pas affichez le message « Produit inexistant » .

5- Dans la classe Pays , ajouter une méthode **« ListerProduit() »** permettant de lister les produits disponibles dans le pays .

6- Créer une Classe **Achat :** ayant comme attributs le produitAcheté : Produit /La quantitéAcheté : entier /et la dateAchat : Date d'aujourd'hui

7- Ajouter une méthode **« MontantAchat() »** permettant de calculer le montant d'un achat (PrixProduit * quantiteAchete)

8- Tester les classes la méthode Main .