

Description

Notre client souhaiterait que l'on réalise une application de gestion bancaire.

Le but est d'obtenir une librairie javascript permettant de lister les informations bancaires, et simuler les retraits, dépôts et calcul des intérêts de ses comptes.

Partie 1

Pour débiter nous avons besoin de 2 class :

- Personne qui aura :
 - Attributs :
 - nom (le nom de la personne)
 - prenom (le prénom de la personne)
 - dateDeNaissance (la date de naissance de la personne)
- Courant, représente un compte courant de notre application, et aura :
 - Attributs :
 - numero (numéro du compte)
 - solde (somme disponible sur le compte, ne peut pas être inférieur à la ligne de crédit)
 - ligneDeCredit (valeur toujours positive représentant la limite négative de notre solde)
 - titulaire (personne à le qui le compte courant appartient)
 - Méthodes :
 - dépôt(montant) (permet d'ajouter un montant positif sur le solde du compte)
 - retrait(montant) (permet de retirer un montant positif sur le solde du compte, attention à ne pas descendre le solde en dessous de la ligne de crédit)

Partie 2

Attention, le solde ne peut être modifié que grâce aux méthodes `depot()` et `retrait()`.

Mettez en place une propriété `Solde` qui donne accès en lecture à l'attribut `solde`.

Une autre propriété `LigneDeCredit`, en lecture et écriture à l'attribut `ligneDeCredit`, qui se chargera de vérifier que celle-ci est toujours positive et non NaN.

Une autre propriété `Titulaire`, en lecture et écriture à l'attribut `titulaire`.

Partie 3

Ensuite nous aurons besoin d'une class représentant la banque :

- Banque aura :
 - Attributs :
 - comptes (tableau de comptes)
 - Propriétés :
 - nom (nom de la banque)
 - Méthodes :
 - compte(numéro) (permet de récupérer un compte à l'aide du numéro de compte de ce dernier)
 - ajouterCompte(compte) (permet d'ajouter un compte à la liste des comptes)
 - avoirsDesComptes(titulaire) (permet de connaître la somme des soldes positifs des comptes appartenant au titulaire)

Partie 4

Les banques ne gèrent pas que des comptes courant, il y a aussi des comptes épargnes.

- Créer la class Epargne :
 - Attributs :
 - numero (numéro du compte)
 - solde (somme disponible sur le compte, ne peut pas être inférieur 0)
 - dateDernierRetrait (correspond à la date du dernier retrait effectué)
 - titulaire (personne à le qui le compte épargne appartient)
 - Propriétés :
 - Solde (lecture seule du solde)
 - DateDernierRetrait (lecture seule de dateDernierRetrait)
 - Titulaire (lecture et écriture de titulaire – doit toujours être de type Personne)
 - Méthodes :
 - depot(montant) (permet d'ajouter un montant positif sur le solde du compte)
 - retrait(montant) (permet de retirer un montant positif sur le solde du compte, attention à ne pas descendre le solde en dessous de 0)

Partie 5

Le grand nombre de similitudes entre Epargne et Courant peut être réduit grâce à l'héritage !
Créez une class Compte, qui serait le point commun entre Epargne et Courant.
Attention, tout n'est pas similaire !

Partie 6

Faite que la Banque puisse traiter les comptes Epargne en plus des comptes Courant.
Ajoutez-y une méthode calculDesInterets() qui aura un comportement spécifique selon le type de Compte :

Epargne : l'intérêt apporte un gain de 3% sur le solde

Courant : l'intérêt apporte un gain de 1,5% sur le solde si ce dernier est positif, sinon un endettement de 4%