

TP 2. METHODES, CONSTRUCTEURS

MESSAGES,

Dans la suite des travaux pratiques, nous allons créer un nouveau projet pour chaque TP. Cela permet de conserver un historique de l'évolution du projet. Pour ce nouveau projet, créer un projet *bankonet-tp02*. Copier les ressources (packages et classes) du projet *bankonet* dans le nouveau projet et finalement fermer le projet *bankonet*.

A chaque nouveau TP, il faut procéder de la même manière pour initialiser le nouveau projet.

EXERCICE 1 DECLARATION DE METHODE

- ⇒ Dans la classe `CompteCourant`, définir les méthodes `crediter(...)` et `debiter(...)` pour gérer les mouvements du compte. Elles prennent en argument le montant à créditer/débité.
- ⇒ `crediter` augmente le solde actuel du montant passé en paramètre
- ⇒ `debiter` diminue le solde actuel du montant passé en paramètre
- ⇒ `toString` renvoie une forme textuelle (les attributs avec du texte de mise en forme)

Nom	Arguments	Type de retour	static
<i>toString</i>		String	non
<i>Débité</i>	double montant	void	non
<i>Créditer</i>	double montant	void	non

Situation actuelle

CompteCourant
String numero String intitule double solde double montantDecouvertAutorise static int nbComptesCourants
public void imprimer(CompteCourant compteCourant) void crediter(double) void debiter(double)

EXERCICE 2 ENVOI DE MESSAGE

- ⇒ Dans la classe `TestCompteCourant`, appeler les méthodes `crediter(...)` et `debiter()` sur `CompteCourant`.



Tester.

EXERCICE 3 CONSTRUCTEURS

- ⇒ Dans la classe `CompteCourant`, ajouter un constructeur prenant en arguments le numéro, l'intitule, le solde, et le découvert autorisé
- ⇒ Modifier la création de comptes dans `TestCompteCourant`. La création doit maintenant passer par l'appel du constructeur « enrichi » de `CompteCourant`.



Tester.

- ⇒ Dans `TestCompteCourant.main(...)`, supprimer les incrémentations de `nbComptesCourants`.
- ⇒ Incrémenter le nombre de comptes courants dans le constructeur par défaut `CompteCourant()`.

- ⇒ Tester l'application. La valeur `nbComptesCourants` n'est pas incrémentée car on n'appelle le constructeur par défaut à aucun moment.
- ⇒ Modifier le constructeur `CompteCourant(...)` pour qu'il prenne en compte l'appel au constructeur par défaut.



Tester.

Situation actuelle

CompteCourant
String numero String intitule double solde double montantDecouvertAutorise static int nbComptesCourants
CompteCourant() CompteCourant(String, String, double, double) void crediter(double) void debiter(double) String toString()

Note : Contrairement à la situation précédente, le constructeur par défaut `CompteCourant()` doit être défini explicitement.