

Volet 1 : Algorithmique

Exercice : Location de vélo

Le but de cet exercice est de permettre à un service de location de vélos (online, tournant 24 heures sur 24) de facturer ses clients.

Le programme demandera à l'utilisateur d'entrer les heures de début et de fin de location sous la forme d'entiers (on ne se préoccupe pas des minutes pour simplifier).

Les tarifs de location sont définis comme suit :

- 10 DA par heure si le vélo est loué entre 0h et 7h ou entre 17h et 24h ;
- 20 DA par heure si le vélo est loué entre 7h et 17h.

Votre programme demandera à l'utilisateur de quelle heure à quelle heure se fait la location et calculera le prix de la location en conséquence.

Vous adopterez les simplifications suivantes :

- les heures de début et fin de location sont des entiers (pas de demi ni de quart, toute heure entamée est due) ;
- l'heure du début de la location est toujours inférieure à l'heure de la fin de la location; cela implique que la location ne peut pas se faire sur plus de 24 heures ; elle doit se faire dans la même journée.

Si les données introduites sont correctes, votre programme affichera simplement le coût de la location en respectant strictement les formats donnés dans les exemples de déroulement ci-dessous. En cas de donnée incorrecte, votre programme devra afficher un message d'erreur et s'arrêter. Utilisez strictement les messages suivants :

- « Les heures doivent être comprises entre 0 et 24 ! » suivi d'un saut de ligne, si une des heures introduites par l'utilisateur n'est pas comprise entre 0 et 24 (inclus) ;
- « Bizarre, vous n'avez pas loué votre vélo bien longtemps ! » suivi d'un saut de ligne, si les heures de début et fin de location sont identiques ;
- et « Bizarre, le début de la location est après la fin ... » suivi d'un saut de ligne si l'heure de début de la location est supérieure à l'heure de fin.

Exemples de déroulement

Il est impératif que votre code respecte le format de réponse suivant :

1) Exemple où la durée de location implique les deux tarifs :

L'heure de début de la location (un entier) = 10

L'heure de fin de la location (un entier) = 19

Résultat à afficher :

Vous avez loué votre vélo pendant

2 heure(s) au tarif horaire de 10 DA

7 heure(s) au tarif horaire de 20 DA

Le montant total à payer est de 160 DA.

2) Exemple où la durée de location n'implique qu'un seul tarif :

L'heure de début de la location (un entier) = 18

L'heure de fin de la location (un entier) = 20

Résultat à afficher :

Vous avez loué votre vélo pendant

2 heure(s) au tarif horaire de 10 DA

Le montant total à payer est de 20 DA.

2) Exemple de données invalides :

L'heure de début de la location (un entier) = 18

L'heure de fin de la location (un entier) = 16

Résultat à afficher :

Bizarre, le début de la location est après la fin ...

Volet 2 : Orienté Objet

Exercice 1 :

Voici le texte d'une classe représentant de façon sommaire un compte bancaire et les opérations bancaires courantes.

```
class Compte{
    int solde = 0;
    void deposer(int montant){
        solde = solde + montant;
    }
    void retirer(int montant){
        solde = solde -montant;
    }
    void virerVers(int montant, Compte destination){
        this.retirer(montant);
        destination.deposer(montant);
    }
    void afficher(){
        System.out.println("solde: "+ solde);
    }
}
```

1. Comment fonctionne la méthode virerVers ?
2. Créez deux comptes que vous affecterez à deux variables. Ecrivez le code correspondant aux opérations suivantes :
 - dépôt de 5.000 Dinars sur le premier compte.
 - dépôt de 10.000 Dinars sur le second compte.
 - retrait de 1.000 Dinars sur le second compte.
 - virement de 1.500 Dinars du premier compte vers le second.
 - affichage des soldes des deux comptes.

Vous mettrez le code java correspondant à cette question dans la méthode main d'une nouvelle classe appelée TesteCompte. Donnez le résultat généré par l'exécution de ce programme.

Exercice 2 :

```
class Compteur {
    int x;
    Compteur(int n) {
        x = n;
    }
    Compteur incremente() {
        x++;
        return this;
    }
    int value() {
        return x;
    }
    public static void main(String[] args) {
        Compteur c1, c2, c3;
        c1 = new Compteur(0);
        c1.incremente();
        c2 = new Compteur(1);
        c3 = c1;
        if (c1 == c3) {
            System.out.println("c1 et c3 sont egaux");
        } else {
            System.out.println("c1 et c3 ne sont pas egaux");
        }
        if (c1.value() == c2.value()) {
            System.out.println("c1 et c2 ont meme valeur");
        } else {
            System.out.println("c1 et c2 n'ont pas la meme valeur");
        }
        if (c1 == c2) {
            System.out.println("c1 et c2 sont egaux");
        } else {
            System.out.println("c1 et c2 ne sont pas egaux");
        }
        if (c1.value() == c1.incremente().value()) {
            System.out.println("c1 et c1 incremente ont meme valeur");
        } else {
            System.out.println("c1 et c1 incremente n'ont pas la meme valeur");
        }
        if (c1 == c1.incremente()) {
            System.out.println("c1 et c1 incremente sont egaux");
        } else {
            System.out.println("c1 et c1 incremente ne sont pas egaux");
        }
    }
}
```

Essayez de prédire le résultat de l'exécution de ce programme. Testez le programme. Que peut-on en déduire sur la notion d'égalité d'objets en java ?