Exercices POO dans dart Série 01

1. Exercice 01

- **Énoncé**
  - Créez une classe appelée **Point** permettant de manipuler un point d'un plan (**x,y**). Vous fournirez :
    - Un constructeur recevant en argument les coordonnées (double) d'un point.
    - Une méthode membre "move" effectuant une translation définie par ses deux arguments (double).
    - une méthode membre "display" affichant simplement les coordonnées du point.
  - Les coordonnées du point seront des données privées membres.
  - Nous écrirons séparément :
    - Un fichier source constituant la déclaration de la classe.
    - Un petit programme de test (principal) déclarant un point, l'affichant, le déplaçant et l'affichant à nouveau.
- **Solution**
  - Essayez de faire l'exercice de votre côté avant de regarder la Solution !

2. Exercice 02

- **Énoncé**
  - Voici le texte d'une classe représentant de façon sommaire un compte bancaire et les opérations bancaires courantes.
    - **Question 1**: Comment fonctionne la méthode virement ? Combien de comptes fait-elle intervenir ?
    - **Question 2**: Créez deux comptes que vous affecterez à deux variables. Écrivez le code correspondant aux opérations suivantes :
      - dépôt de 500 euros sur le premier compte.
      - dépôt de 1000 euros sur le second compte.
      - retrait de 10 euros sur le second compte.
      - virement de 75 euros du premier compte vers le second.

apcpedagogie.com

- affichage des soldes des deux comptes.
- Vous mettrez le code Dart correspondant à cette question dans la méthode **main**. Vous compilerez et testerez ce programme.

- **Question 3**: Créez un tableau de dix comptes. Pour cela, notez bien qu'il faut d'abord créer le tableau puis créer successivement les dix comptes à mettre dans les dix cases de ce tableau. Dans chaque case, faites un dépot de 200 euros plus une somme égale à 100 fois l'indice du compte dans le tableau. Ensuite, vous ferez un virement de 20 euros de chaque compte vers chacun des comptes qui le suivent dans le tableau (par exemple, du compte d'indice 5, il faut faire des virements vers les comptes d'indice 6, 7, 8 et 9).

```
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(){
Terminal.ecrireString("solde: "+ solde);
}
}
```

- **Solution**

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

3. Exercice 03

- **Énoncé**

- Cet exercice reprend la classe **Compte** de l'exercice précédent(exercice02).
- **Question 1**
  - Complétez la classe Compte avec une information supplémentaire : le nom du titulaire du compte (type String). Vous modifierez la méthode d'affichage

apcpedagogie.com

pour qu'elle affiche cette information.

- **Question 2**
  - Créez un constructeur pour la classe Compte. Ce constructeur doit prendre en paramètre le nom du titulaire du compte.
  - Donnez le code de création d'un compte qui appelle ce constructeur.
- **Question 3**
  - Faut-il prévoir des méthodes permettant de changer le nom du titulaire du compte ?

- **Solution**

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

4. Exercice 04

1. **Énoncé**

- Soit le programme suivant :
- Refactorisez ce code en créant une classe **Person** qui contiendra les propriétés name, age, height.
- Cette classe devrait avoir une méthode **printDescription()** qui peut être utilisée pour imprimer le nom, l'âge et la taille comme dans le programme ci-contre.
- Une fois cela fait, créez deux instances de **Person** et utilisez-les pour appeler la méthode **printDescription()** que vous venez de créer.

```
void main() {
  final person = {
    'name': 'Mohamed',
    'age': 36,
    'height': 1.84
  };
  print("Je m'appelle ${person['name']}
.J'ai ${person['age']} ans,
 je mesure ${person['height']} mètres.");
}
```

2. **Solution**

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

apcpedagogie.com

5. Exercice 05

1. Énoncé

- On a pour objectif de développer un **programme en Dart** pour gérer les coûts des enseignants d'une université.
- Chaque enseignant est caractérisé par son nom, prénom et le nombre d'heures de cours qu'il assure dans l'année. Il existe plusieurs catégories d'enseignants :
  - Les enseignants-chercheurs, qui reçoivent un salaire fixe mensuel ainsi qu'une rémunération pour les heures complémentaires.
  - Les vacataires, recrutés par des organismes externes à l'université, qui sont payés un tarif horaire fixe.
  - Les doctorants, qui peuvent donner des cours jusqu'à un certain nombre d'heures et sont rémunérés selon un tarif horaire.
  - Les enseignants-chercheurs reçoivent un salaire mensuel fixe et un montant supplémentaire pour les heures complémentaires au-delà d'un certain seuil. Les vacataires perçoivent un tarif horaire fixe pour toutes leurs heures. Les doctorants sont payés un tarif horaire fixe, mais leur nombre d'heures de cours est limité.
  - Pour chaque enseignant, le programme doit calculer le coût total pour l'université, en tenant compte du salaire et des charges applicables sur ce salaire.
  - Nous allons élaborer une solution en Dart qui respecte les principes de la programmation orientée objet pour modéliser ces différentes catégories d'enseignants et calculer leurs coûts respectifs.

2. Solution

- Essayez de faire l'exercice de votre côté avant de regarder la Solution !

Cours Dart

Source : https://apcpedagogie.com/exercices-poo-dans-dart-serie-01 · téléchargé le 07/11/2026 · © APC Pédagogie · apcpedagogie.com