
Exercice 1

Écrivez une classe représentant une ville. Elle doit avoir les propriétés nom et département et une méthode affichant « la ville X est dans le département Y ». Créez des objets ville, affectez leurs propriétés, et utilisez la méthode d'affichage.

Exercice 2

Modifiez la classe précédente en la dotant d'un constructeur. Réalisez les mêmes opérations de création d'objets et d'affichage.

Exercice 3

Créez une classe représentant une personne. Elle doit avoir les propriétés nom, prénom et adresse, ainsi qu'un constructeur et un destructeur. Une méthode `getpersonne()` doit retourner les coordonnées complètes de la personne. Une méthode `setadresse()` doit permettre de modifier l'adresse de la personne. Créez des objets personne, et utilisez l'ensemble des méthodes.

Exercice 4

Créez une classe nommée `form` représentant un formulaire HTML. Le constructeur doit créer le code d'en-tête du formulaire en utilisant les éléments `<form>` et `<fieldset>`. Une méthode `settext()` doit permettre d'ajouter une zone de texte. Une méthode `setsubmit()` doit permettre d'ajouter un bouton d'envoi. Les paramètres de ces méthodes doivent correspondre aux attributs des éléments HTML correspondants. La méthode `getform()` doit retourner tout le code HTML de création du formulaire. Créez des objets `form`, et ajoutez-y deux zones de texte et un bouton d'envoi. Testez l'affichage obtenu. Le fichier contenant la définition de la classe `form` (`ch9exo4.php`) est indépendant ce qui permet son inclusion dans d'autres scripts en vue de l'utilisation de la classe ou de son extension.

Exercice 5

Créez une sous-classe nommée `form2` en dérivant la classe `form` de l'exercice 4. Cette nouvelle classe doit permettre de créer des formulaires ayant en plus des boutons radio et des cases à cocher. Elle doit donc avoir les méthodes supplémentaires qui correspondent à ces créations. Créez des objets, et testez le résultat.

Exercice 7

Créez une classe abstraite représentant une personne. Elle déclare les propriétés nom et prénom et un constructeur. Créez une classe client dérivée de la classe personne en y ajoutant la propriété adresse et une méthode setcoord() qui affiche les coordonnées complètes de la personne. Créez une classe électeur dérivée de la même classe abstraite, et ajoutez-y deux propriétés bureau_de_vote et vote, ainsi qu'une méthode avoter(), qui enregistre si une personne a voté dans la propriété vote. Les propriétés nom et prénom qui sont déclarées private dans la classe abstraite (repères 1 et 2) sont redéfinies comme public dans la classe electeur (repères 3 et 4) car sinon nous ne pourrions pas les lire lors du contrôle du vote(repères 5 et 6) car la classe electeur ne possède pas de méthode getinfo().

Exercice 8

Exercice 8 Créez deux namespaces nommés Firme::Client et Firme::Commercial possédant chacun des classes Personne. Chacune d'elles doit avoir des propriétés pour enregistrer les coordonnées de la personne et son code, un constructeur, des méthodes set() et get() pour pouvoir modifier et afficher les propriétés. Créez ensuite des objets représentant deux clients et un commercial. Création du namespace Firme\Client fichier chap9exo8.php
NB1: la notation est celle de 5.3 antislash \ entre Firme et Client

Exercice 9

Intégrez les méthodes magiques connues à au moins une des classes de l'exercice 8 après avoir déclaré la propriété code comme protected. Création du namespace Firme\Client fichier chap9exo9.php