

Web Avancé

TP N°5 : PHP –Les bases-

Exercice 1 : Années bissextiles

Créer une fonction en PHP qui vérifie si une année entrée par un l'utilisateur via un formulaire bissextile et qui affiche si l'année entrée est bissextile ou non. Une année est bissextile si elle est divisible par 4, mais pas par 100. Les années divisibles par 400 sont cependant bissextiles.

Testez votre programme avec les années 1900, 1984, 2010 et 2020.

Exercice 2 : Traitement des données de formulaires

1. Créer un formulaire de coordonnées personnelles composé des champs suivants : Nom complet du client, l'email, le téléphone, l'adresse, et code postal. Les données sont ensuite traitées par un fichier PHP séparé récupérant les données et les affichant dans un tableau HTML.

2. Améliorer le script de traitement en vérifiant ce qui suit :

- a) Tous les champs sont obligatoires
- b) Le champ email doit contenir le caractère @.
- c) Le champ téléphone doit être sous la forme (06 ou 05) 00 00 00 00.
- d) Le code postal doit contenir 5 chiffres seulement.

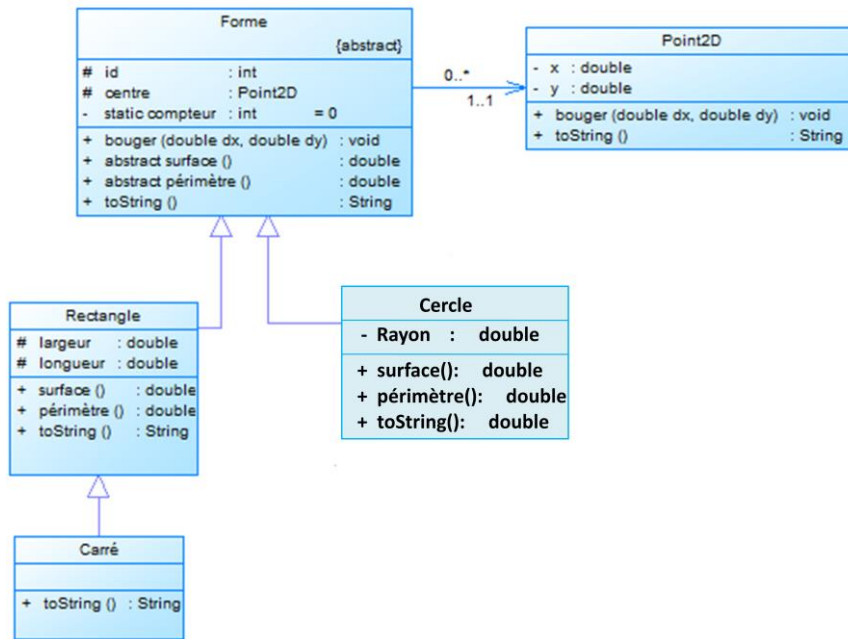
3. Comment faire en sorte que les données soient traitées par le même fichier que celui qui contient le formulaire.

Exercice 3 : Gérer des objets 1

1. Créer une classe abstraite représentant une personne. Elle déclare les propriétés nom et prénom et un constructeur.
2. 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.
3. 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.
4. Tester votre code avec des instances de types client et électeur

Exercice 4 : Gérer des objets 2

On souhaite créer un ensemble de classes permettant de manipuler des formes. Pour cela on propose le diagramme de classes suivant :



I. La classe Point2D en php :

1. Créer la classe Point2D caractérisée par 2 coordonnées x ,y.
2. Définir les accesseurs et le constructeur d'initialisation.
3. Ajouter à la classe la fonction magique __toString qui permet d'afficher les propriétés d'un objet Point2D sous la forme : **Point(x =10,y =3)**
4. Ajouter à la classe une méthode bouger prenant en paramètre les deux réels dx et dy et qui permet de translater le point vers le point x+dx ,y+dy et z+dz.

II. La classe abstraite Forme en php :

1. La classe Forme possède un centre de type Point2D et un id.
2. Lors de la création d'une Forme , son id est incrémenté selon le nombre de Formes créés.
3. Ajouter à la classe les deux méthodes abstraites surface() et périmètre().
4. Ajouter la méthode bouger() qui fait tout simplement appelle à la méthode bouger() de son centre.

III. La classe Rectangle en php :

1. Créer la classe Rectangle dérivée de la classe Forme et caractérisée par deux attributs : largeur et longueur.
2. Ajouter les accesseurs et un constructeur d'initialisation à la classe.
3. Définir la méthode surface qui renvoie la surface d'un objet rectangle : largeur*longueur.
4. Définir la méthode périmètre qui renvoie le périmètre d'un objet rectangle : 2*(largeur+longueur).
5. La méthode __toString() représente chaque rectangle sous la forme :

```

{Rectangle :1          //1 est l'id de la forme
    Centre:Point(x=2, y=3)
    Largeur:1
    Longueur:2
    surface:2
  
```

```

    perimetre:6
}

```

IV. La classe Cercle :

1. Créer la classe Cercle qui hérite de la classe Forme et qui possède en plus un rayon.
2. Ajouter les accesseurs et un constructeur d'initialisation .
3. Définir la méthode surface qui renvoie la surface d'un objet cercle : $\pi * r^2$
4. Définir la méthode périmètre qui renvoie le périmètre d'un objet rectangle : $2 * \pi * r$
5. La méthode __toString() représente chaque cercle sous la forme :

```

{Cercle :2
  Centre:Point(x=3, y=3)
  Rayon:2
  surface:12.57
  perimetre:12.57
}

```

V. La classe Carré :

1. Créer la classe Carré qui hérite de la classe Rectangle (un carré est un rectangle avec largeur = longueur).
2. La classe Carré ne pourra pas être dérivée.
3. Attribuer à la classe un constructeur d'initialisation .
4. La méthode __toString() représente chaque carré sous la forme :

```

{Carre :3
  Centre:Point(x=3, y=3)
  Longueur:1
  surface:1
  perimetre:4
}

```

VI. Ecrire un script de Test :

```

{Rectangle :1
  Centre:Point(x=2, y=3)
  Largeur:1
  Longueur:2
  surface:2
  perimetre:6
}

```

```

{Cercle :2
  Centre:Point(x=1, y=2)
  Rayon:2
  surface:12.57
  perimetre:12.57
}

```

```
//Après déplacement du cercle par (dx=2,dy=1)
```

```
{Cercle :2
  Centre:Point(x=3, y=3)
  Rayon:2
  surface:12.57
  perimetre:12.57
}
```

```
{Carre :3
  Centre:Point(x=3, y=3)
  Longueur:1
  surface:1
  perimetre:4
}
```