Travaux Pratiques
R5.05 Développement Front Avancé.

Exercice 1 - Classes JavaScript ES6

Pour cet exercice, on utilisera les classes JavaScript (mot-clé class) définies dans ES6.

1. Ecrivez une classe Personne qui doit comporter :
— un constructeur qui prendra en paramètre un nom et un age
— une méthode sePresenter() qui affiche dans la console "Je suis (nom) et j'ai (age) ans".

2. Créez maintenant une classe Enfant qui hérite de la classe Personne et qui :
— redéfinit la méthode sePresenter() pour ajouter à la présentation initiale "Je suis un enfant".

3. Complétez maintenant la classe Personne afin de :
— compter le nombre d'instances de personnes créées (personnes et enfants).
— afficher dans la console ce nombre d'instances lorsqu'on appelle la méthode afficherNombrePersonnes().

5. Pour tester votre code, vous devez :
— créer au moins une instance de Personne et d'Enfant,
— appeler la méthode sePrésenter() pour chacune de ces instances,
— appeler la méthode afficherNombrePersonnes().

Exercice 2 - TypeScript

Pour cet exercice, vous utiliserez le langage TypeScript et vous transcompilerez le code en ES6. Il vous faudra donc créer un nouveau projet TypeScript et utiliser un fichier de configuration tsconfig.json.

Vous utiliserez les classes TypeScript (mot-clé class) sans implémenter d'interface. Toutes les bonnes pratiques de TypeScript doivent être mises en oeuvre (déclaration des types des variables, déclaration des types de retour des fonctions, déclaration des propriétés (private/public/protected), etc...

1. Ecrivez une classe Point ayant les propriétés suivantes :
— abs : une propriété privée de type number, qui correspondra à l'abscisse du point,
— ord : une propriété privée de type number, qui correspondra à l'ordonnée du point.

2. Créez le constructeur de cette classe qui nécessitera deux paramètres de type number (l'abscisse et l'ordonnée) du point à créer.

3. Créez la méthode sePresenter() qui ne retourne rien mais qui affiche dans la console "‘Mon abscisse est (valeur de l'abscisse) et mon ordonnée (valeur de l'ordonnée)".

4. Ecrivez la méthode calculerDistance(p : Point) qui renvoie la distance entre le point de l'objet courant (this) et l'objet Point p passé en paramètre. Créez ensuite deux instances de la classe Point et affichez la distance entre ces deux points dans la console.

5. Écrivez la méthode calculerMilieu(p : Point) qui permet de calculer et retourner un objet correspondant au milieu du segment défini par le point de l'objet courant (this) et l'objet Point p passé en paramètre.

6. Créez maintenant une classe TroisPoints ayant les propriétés suivantes ainsi que son constructeur :
- point1 : une propriété privée de type Point, qui correspondra au premier point,
- point2 : une propriété privée de type Point, qui correspondra au deuxième point,
- point3 : une propriété privée de type Point, qui correspondra au troisième point.

7. Ecrivez une méthode estEquilateral() qui retourne true si les trois points forment un triangle équilatéral, false sinon. Testez votre code pour les différents cas de figure. Vous afficherez le résultat dans la console.