

Sujet 3 complémentaire

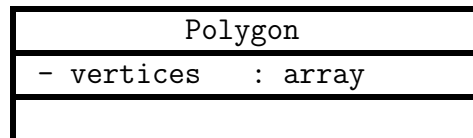
Avant de commencer

Vous stockerez dans le répertoire TP3 les scripts *Polygon.php* et *TestPolygon.php* dans lesquels se trouvera l'ensemble des développements produits tout au long de ce TP complémentaire. N'oubliez pas de mettre à jour systématiquement votre dépôt git.

La classe Polygone

Dans cet exercice on utilisera la classe la classe `Point` implémentée dans l'exercice 3 du TP3. la documentation de cette classe est OBLIGATOIRE.

Un polygone peut-être décrit grâce à l'ensemble de ses sommets (points). Un objet de type `Polygon` peut-être alors être assimilé à un tableau ordonné de sommets `vertices`. Ci-dessous vous trouverez le diagramme de classe correspondant :



Question 1.

Dans la classe `Polygon`, on souhaite disposer d'un constructeur qui prendra en paramètres une tableau de points qui servira à initialiser l'attribut d'instance `vertices`. Proposez une définition de ce constructeur en veillant à respecter le principe d'encapsulation des données et testez-le.

Question 2.

Dans la classe `Polygon`, on souhaite disposer d'une méthode d'instance `getVerticesNumber`. Cette méthode retournera le nombre de sommets que contient un polygone. Définissez cette méthode et testez-la.

Question 3.

Dans la classe `Polygon`, on souhaite disposer d'une méthode d'instance `isEmpty`. Cette méthode retourne `true` si le nombre de sommets du polygone est nul, `false` sinon. Définissez cette méthode et testez-la.

Question 4.

Dans la classe `Polygon`, on souhaite disposer d'une méthode d'instance `getPerimeter` qui retourne le périmètre d'un polygone, c'est-à-dire la somme des longueurs de ses arrêtes. Définissez cette méthode et testez-la.

Question 5.

Dans la classe `Polygon`, on souhaite disposer d'une méthode d'instance `insertPoint` qui prend en paramètre un nouveau point ainsi que la valeur d'un *indice*. Cette méthode permet d'insérer

le point dans le tableau **vertices** à la position *indice*. Si l'insertion est impossible (*indice* incorrect), la méthode retournera *false*, *true* sinon. Définissez cette méthode et testez-la.

Question 6.

En vous inspirant du rendu demandé pour un segment, définissez la méthode `_ _toString ()` dans la classe **Polygon** et testez-la.

Question 7.

Sachant que deux polygones sont égaux si les points les définissant sont égaux, définissez la méthode d'instance **isEqual** dans la classe **Polygon**. Attention, deux polygones pour lesquels le point de départ n'est pas placé au même endroit dans l'attribut d'instance **vertices** mais dont l'ordre des points est respecté au sens près sont considérés comme égaux.