Laboratoire 08

Programmation Objet 1

420-620-MA

Programmation en langage de haut niveau

Nom :…………………………………………

* **Remettre votre projet via Léa**

Programmation Objet 1

**Étape 1 - Création de la classe Rectangle**

1. Créez une application console.
2. Ajouter une nouvelle classe nommée Rectangle dans un fichier Rectangle.cs.
3. Ajouter un nouvel attribut public nommé « hauteur » de type int.
4. Ajouter un nouvel attribut public nommé « largeur » de type int.
5. Ajouter une méthode publique nommée « Perimetre » qui retourne le périmètre du rectangle.
6. Ajouter une méthode publique « Aire » qui calcul l’aire du rectangle.
7. Attention! Les méthodes des points 5 et 6 ne doivent rien afficher!!!
8. Dans le fichier Program.cs, créez une méthode nommée Version1. Dans cette méthode, créez un nouvel objet rect1 de dimensions 10 x 30, et un autre rectangle de dimensions 15 x 28.
9. Affichez, pour chacun des rectangles, l’aire et le périmètre, en appelant les méthodes d’instances, toujours dans la méthode Version1.
10. Appelez Version1 à partir de Main et testez que le tout fonctionne.

**Étape 02 : Utilisations d’accesseurs**

1. Faites maintenant une classe RectangleV2 dans un fichier à part. (Copiez-collez la classe rectangle de l’étape précédente pour aller plus vite! Nous allons modifier cette classe! Nous conservons la V1 afin de voir les différences entre les deux versions.)
2. Dans RectangleV2, rendez les attributs hauteur et largeur privés.
3. Ajouter les accesseur (getter et setter) de l’attribut hauteur.
4. Ajouter les accesseur (getter et setter) de l’attribut largeur.
5. On calcule maintenant le périmètre et l’aire à l’aide maintenant d’un accesseur en lecture seule. Créez donc deux accesseurs en lecture seule pour le périmètre et l’aire et supprimez les méthodes de calcul d’aire et de périmètre.
6. Dans Program.cs, créez une méthode Version2, copier-coller de Version1.
7. Dans cette méthode, au lieu d’utiliser la classe Rectangle (V1), utilisez la classe RectangleV2.
8. Modifiez la façon de spécifier la hauteur et la largeur de chacun des rectangles.
9. Modifiez les appels faits pour l’affichage de l’aire et du périmètre de chaque rectangle.
10. Faites appel à la méthode Version2 à partir de Main (vous pouvez commenter Version1 pour faciliter les tests).

**Étape 03 : Constructeurs**

1. Faites une classe RectangleV3 basée sur RectangleV2 (copier-coller), de la même manière que nous avons fait RectangleV2 à partir de Rectangle.
2. Dans cette 3ième version, les accesseurs hauteur et largeur sont en lecture seule. Comment allons-nous alors spécifier la hauteur et la largeur? Par un constructeur.
3. Dans Program.cs, faites une méthode Version3 basée sur version2.
4. Adaptez le code de la version 3 pour faire appel à RectangleV3, et ajustez les appels en conséquence pour que le tout fonctionne comme avant!
5. Lorsqu’un rectangle est crée, on doit afficher un message « Un nouveau rectangle vient d’être crée! ». Cependant, ce message doit s’afficher à partir de la classe RectangleV3.
6. Ajoutez maintenant un deuxième constructeur qui ne prend qu’un seul paramètre (int dimension), qui crée alors un rectangle carré dont la hauteur sera égale à la largeur.
7. Dans Version3, créez un troisième rectangle (cette fois-ci un carré) de dimension 50, à partir du constructeur nouvellement codé.
8. Affichez l’aire et le périmètre de ce dernier carré(rectangle).

**Étape 04 : Méthode toString()**

1. Créez une version 4 de Rectangle! (À partir de RectangleV3).
2. Créez une méthode (dans RectangleV4) toString, qui retourne une chaîne de caractère caractérisant le rectangle en question. Par exemple :

Rectangle 34 (h) x 28 (l).

1. Dans la méthode Version4 (que vous aurez préalablement copiée de Version3), avant d’afficher le périmètre et l’aire des trois objets RectangleV4, appelez la méthode toString() et affichez ce que cette méthode vous retourne.

**Étape 05 : Membre statique**

1. Créez une version 5 de Rectangle! (À partir de RectangleV4).
2. Maintenant, on veut modifier le message qui s’affiche à la création afin qu’il affiche le nombre total de rectangles crée à présent. Par exemple, lorsque le rectangle-carré sera crée, on affichera :

«Un nouveau rectangle vient d’être crée. Ceci est l’instance #3».

Pour ce faire, il faut passer maintenant par un membre statique. Le membre statique est une donnée ou méthode qui s’applique à la classe (l’ensemble des instances) et non une instance en particulier.

1. Créez donc un attribut (variable) nbInstances de type Int, initialisé à zéro.
2. Dans les constructeurs, incrémentez cette variable de un pour qu’à chaque fois qu’un rectangle est créé, qu’alors cette variable soit incrémentée.
3. Modifiez le message en conséquence.
4. Finalement, faites une méthode Version5 basée sur Version4 qui teste le tout.
5. Au final, laissez les appels Version1, Version2, etc. dans la méthode Main.