

## 420-3GP-BB

### Module 1 – Introduction à C# (Exercices)

#### Exercice 1

Écrire une classe permettant de décrire un livre et de positionner les valeurs associées.

Voici la **liste des propriétés**, ou attributs, d'un livre :

- Titre
- Auteur
- Éditeur
- Nombre de pages (Doit être plus grand que 0).
- Année

La **liste des méthodes** permettant de gérer ces propriétés:

- Accéder aux attributs (get).
- Faites un constructeur qui prend toutes les valeurs nécessaires.
- Créer un nouvel objet livre via un constructeur prenant les différentes valeurs pour le livre
- Implémentez la méthode ToString() qui retourne une chaîne contenant les informations sur le livre. Pour construire la chaîne, utiliser un objet de la classe StringBuilder.

Dans votre programme principal, créez cinq livres et affichez les informations. Les livres doivent être stockés dans un tableau. Si le nombre de page n'est pas plus grand que 0, lancez l'exception ArgumentException.

Voici à quoi devrait ressembler votre affichage :

```
Game Engine Architecture 3rd ed., Jason Grogory, CRC Press, 1200 pages, 2018.  
Python for Data Analysis 2nd ed., Wes McKinney, O'Reilly, 523 pages, 2017.  
Fundamentals of Game Design 3rd ed., Ernest Adams, News Rider, 556 pages, 2014.  
Game Programming Patterns, Robert Nystrom, genever benning, 345 pages, 2014.  
Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow 2nd ed., Aurélien Géron, O'Reilly, 814 pages, 2019.
```

## Exercice 2

En utilisant les classes, l'encapsulation et l'héritage, créez un programme qui permet de calculer le périmètre d'un rectangle, d'un cercle et d'un carré. Vous aurez donc les classes Carre, Rectangle et Cercle.

Les classes Rectangle et Cercle doivent hériter de la classe abstraite FormeGeometrique qui contiendra une méthode abstraite CalculerPerimetre(). La classe Carre doit hériter de la classe Rectangle.

Les propriétés des formes doivent avoir des accesseurs (get) publics et des mutateurs (set) privés. On doit s'assurer que les dimensions sont plus grandes que 0.0. Sinon, on lance l'exception de type ArgumentException. Les dimensions devront être des doubles.

Les constructeurs doivent prendre toutes les valeurs requises pour l'objet en question.

Testez ensuite votre programme avec trois instances de FormeGeometrique.

Testez également le lancement de votre exception dans un bloc try ... catch.

Voici un exemple d'exécution :

```
Le périmètre du rectangle est : 13,6  
Le périmètre du cercle est : 20,734511513692635  
Le périmètre du carré est : 10  
Valeur non positive
```

### Exercice 3

Créez une interface IDessinable qui aura comme méthode DessinerObjet(). Faites en sorte que la classe Rectangle implémente cette interface.

Puisque la largeur et la hauteur sont des doubles, prenez le plafond pour dessiner les rectangles. Testez ensuite avec un carré et un rectangle.

Voici le résultat possible :

```
Voici le rectangle:
***
***
***
***
***

Voici le carre:
***
***
***
```

### Exercice 4

Revenons à l'exercice 1.

Faites en sorte que la classe Livre implémente l'interface IComparable. Implémentez ensuite la méthode CompareTo afin de pouvoir trier en ordre alphabétique les livres avec la méthode Array.Sort.

Voici le résultat :

```
Fundamentals of Game Design 3rd ed., Ernest Adams, News Rider, 556 pages, 2014.
Game Engine Architecture 3rd ed., Jason Grogory, CRC Press, 1200 pages, 2018.
Game Programming Patterns, Robert Nystrom, genever benning, 345 pages, 2014.
Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow 2nd ed., Aurélien Géron, O'Reilly, 814 pages, 2019.
Python for Data Analysis 2nd ed., Wes McKinney, O'Reilly, 523 pages, 2017.
```

Dans le prochain module, nous verrons comment on peut créer divers projets réutilisant les mêmes classes sans avoir à copier/coller les classes communes.