

Travaux pratiques C#

Série d'exercices 3

Exercice 1

On souhaite gérer un historique de 5 valeurs successives d'une variable de type entier. Pour cela, dans la classe principale, déclarer un tableau de 5 entiers, puis une fonction statique qui sauvegarde une valeur entière dans le tableau, et une fonction statique qui retourne la dernière valeur sauvegardée dans le tableau.

Se servir de ces 2 fonctions pour sauvegarder les valeurs successives d'une variable.

Exercice 2

On travaille maintenant avec des points. La classe Point définit 2 coordonnées entières, un constructeur à 2 arguments entiers, une sur-définition de la fonction ToString et une propriété X qui permet de manipuler la première coordonnée.

On souhaite gérer une seule valeur d'historique de chaque Point : pour cela déclarer dans la classe Point une variable hist de type Point, une fonction sauv qui affecte l'historique à la valeur courante du Point, et restaure qui restitue la valeur précédente d'un Point.

Dans la fonction Main, créez un Point, sauvegardez sa valeur actuelle, modifiez sa première coordonnée, affichez-le puis restaurez sa version précédente.

Dans une première version, on pourra construire un nouveau Point (ajouter un message dans le constructeur pour confirmer la construction d'une nouvelle instance).

Corrigez ensuite en créant un clone avec la fonction MemberwiseClone.

Exercice 3

On travaille maintenant avec une classe Disque. Un disque est composé d'un Point, son centre, et d'un rayon (valeur entière).

Proposez un constructeur à 3 arguments entiers, les 2 premiers qui serviront à construire le centre. Sur-définissez la fonction ToString.

Proposez une propriété qui permet d'obtenir le centre d'un Disque (uniquement get).

Proposez un second constructeur qui prend 2 paramètres : un Point et un rayon.

On va maintenant gérer une instance historique d'un Disque, comme dans l'exercice précédent.

Utilisez dans un premier temps la fonction MemberwiseClone et constatez.

Corrigez en sur-définissant la fonction Clone.