

# Exercices

## Les basiques

1. Affichez « Hello » et « World » sur 2 lignes séparés
2. Créez 2 fonctions dans votre classe, une pour récupérer l'entrée utilisateur en chaine de caractère et l'autre pour récupérer un entier. Utilisez ces méthodes dans les prochains exercices
3. Faites rentrer 3 mots à l'utilisateur et afficher les dans l'ordre inverse de la saisie
4. Affichez si les 2 nombres rentrés par l'utilisateur sont égaux
5. Créez 2 nombre, les afficher, les intervertir et les réafficher
6. Affichez le résultat de l'ajout, la soustraction, la multiplication et la division de 2 nombres rentrés par l'utilisateur
7. Affichez la moyenne de 4 nombres rentrés par l'utilisateur
8. Affichez le nombre le plus grand de 4 nombres rentrés par l'utilisateur
9. Savoir si le nombre rentré est négatif ou positif
10. Savoir si le nombre donné est pair ou impair
11. Rentre un nombre en Celsius et le convertir en Kelvin et Fahrenheit
12. Rentre un chiffre et dire de quel jour il s'agit sous forme de texte
13. Bloquez un utilisateur après avoir rentré 3 faux password
14. Rentre un nombre, puis un signe (+ - \* /), puis un autre nombre, affichez ensuite le résultat
15. Afficher les 10 premiers chiffres à partir de 0 (avec une boucle for)
16. Donnez la table de multiplication des 10 premiers chiffres
17. Rentre une lettre et une taille et l'afficher sous forme de triangle
18. Faire une pyramide d'une hauteur donnée
19. Faire un diamant d'une hauteur donnée
20. Créez la date d'aujourd'hui et affichez-la de façon complète puis de façon jour/mois/année)
21. Créer une date avec l'utilisateur qui rentre l'année, mois, jour, heure, minute et seconde et afficher le jour (Lundi, mardi...) | en français ou en anglais
22. Affichez la différence de temps entre 2 dates
23. Convertir une chaine de caractère en date
24. Jeu du plus ou moins (un nombre est défini par défaut, puis l'utilisateur rentre un chiffre et le programme lui dit si le bon nombre est + ou - que celui défini jusqu'à que l'utilisateur trouve)
25. Affichez une chaine donnée en majuscule
26. Affichez le nombre de caractère d'une chaine sans utiliser la propriété Length
27. Affichez un espace entre toutes les lettres d'une chaine de caractère donnée en inversant les lettres
28. Dire si une chaine est égale à une autre sans prendre en compte la casse
29. Savoir si la chaine donnée est un palindrome (mot qui se lit dans les deux sens : SOS)
30. Lister les nombres de 1 à 100, en remplaçant le nombre par un mot selon certaines conditions
  - a. Le mot Fizz s'il est multiple de 3
  - b. Le mot Buzz s'il est multiple de 5
  - c. Le mot FizzBuzz s'il est multiple de 3 et de 5
  - d. Un retour à la ligne tous les 9 éléments

Bonus : Créez le jeu du morpion (Numéro 300 sur la correction)

## Les classes

31. Créez une classe Personne qui comporte trois propriétés, nom, prénom et date de naissance
32. Ecrivez une méthode qui inverse le prénom d'une personne
33. Créez une personne avec des valeurs rentrées par l'utilisateur, appelez la méthode créée précédemment et affichez la personne
34. Créez 1 Personne, l'affecter à une autre variable Personne, les afficher, appeler la méthode inverser prénom sur la première personne et les réafficher
35. Créez une propriété statique Test avec une valeur par défaut et l'afficher

## L'héritage, le polymorphisme et les interfaces

36. Classes suivantes :

Créez une interface ITravailleur qui a comme méthode Afficher.

Une classe Personne qui comporte trois propriétés, nom, prénom et date de naissance. Cette classe doit implémenter l'interface ITravailleur.

Cette classe comporte un constructeur pour permettre d'initialiser les données. Elle comporte également une méthode polymorphe Afficher pour afficher les données d'une personne.

Une classe Employé qui dérive de la classe Personne, avec en plus un champ Salaire accompagné de sa propriété, un constructeur et la redéfinition de la méthode Afficher.

Réécrivez les méthodes ToString()

Les 2 classes devront implémenter leurs versions de Afficher.

- Ecrire les classes Personne et Employé.
- Créez un programme de test qui comporte tableau de 4 personnes avec 2 personnes et deux employés.
- Affichez l'ensemble des éléments du tableau à l'aide de foreach et de la méthode afficher.
- Faire le même test avec un tableau de 4 ITravailleur.

Bonus : exporter vos classes dans une DLL à part.

## Les tableaux et listes

1. Créez un tableau d'entier, trier le tableau et afficher tous les nombres dans l'ordre
2. Affichez une liste de string par ordre alphabétique
3. Afficher une liste de personne
4. Afficher un dictionnaire de personne avec un entier Id en clé
5. Tester le Contains avec le Equals et IEquatable
6. Trié la liste de personne avec IComparable puis IComparer sur le nom puis le prénom

Bonus : tester les Stacks et Queue

## Linq

Sur une liste d'entier :

1. Afficher la somme

2. Afficher la moyenne
3. Sélectionner que les nombres qui sont > 6
4. Compter le nombre d'élément

Sur une liste de personne :

5. Les trier avec la méthode OrderBy par le nom puis le prénom
6. Afficher que les personnes qui ont leurs noms qui commence par D
7. Afficher que les noms des personnes
8. Afficher le nom en majuscule et le prénom des personnes trié par ordre descendant sur le prénom dont leurs noms commence par D
9. Sélectionner que les 2 première personne
10. Sélection les personnes rentrés en 7<sup>ème</sup> 8<sup>ème</sup> et 9<sup>ème</sup> position
11. Récupérer la personne qui a le nom Dupont
12. Savoir si notre liste contient une personne dont le nom commence par D
13. Regrouper les personnes par leurs prénoms

## Les exceptions

7. Gérer la division par 0 par un try catch
8. Créer une classe Élèves caractérisée par nom, âge et moyenne.
  - a. L'âge doit être entre 18 et 26 sinon l'exception InvalidAgeException (elle affiche le message "L'âge doit être entre 18 et 26") est générée.
  - b. La note doit être entre 0 et 20 sinon l'exception InvalidNoteException est générée (elle affiche le message "La note doit être entre 0 et 20").
  - c. Définir les constructeurs de la classe, les accesseurs

## Les tests unitaires

1. Rajouter des tests dans votre application

## Aspects Avancés

57. Généricité
  - a. Créer une nouvelle classe « MyPair » qui a deux types génériques T et U.
  - b. Cette classe a deux propriétés génériques First et Second et un constructeur générique.
  - c. Ecrire la méthode ToString pour avoir le résultat suivant : « Value 1 : First – Value 2 : Second »
  - d. Créer une instance de MyPair <String, int> et l'afficher.
58. Delegate
  - a. Créer une nouvelle classe « Exercice ».
  - b. Créer deux delegates, un retournant void et un autre avec deux paramètres de type double.
  - c. Créer une méthode statique « Welcome » qui affiche un texte sur la console et une autre « Add » qui prend deux doubles paramètres et affiche leur somme.
  - d. Utilisez les delegates pour exécuter les 2 méthodes
59. Action
  - a. Créez une méthode DoSomething qui a un paramètre callback (Action) qui est une fonction avec 1 paramètre de type string. Appelez la méthode DoSomething avec l'implémentation du callback avec un writeLine à l'intérieur.

## 60. Attribut

- a. Tester l'attribut :

```
[System.Obsolete("use NewMethod", true)]
```

## Thread

Tester les exemples du cours

## IO

Sérialiser et désérialiser une liste de personnes dans un fichier