

# QCM développement en C#

---

1. Le mot clef `int` correspond à quel type .NET ?

*1 seule réponse possible*

- ☐ `System.Int16`
- ☒ `System.Int32`
- ☐ `System.Int64`
- ☐ `System.Int128`

2. Laquelle de ces définitions de `string` évite le recours aux séquences d'échappement en C# ?

*1 seule réponse possible*

- ☐ `string s = #"Test string";`
- ☐ `string s = "Test string";`
- ☒ `string s = @"Test string";`
- ☐ `string s = "Test string";`

3. Comment déclare-t-on correctement un tableau à deux dimensions en C# ?

*1 seule réponse possible*

- ☒ `int[,] myArray;`
- ☐ `int[][] myArray;`
- ☐ `int[2] myArray;`
- ☐ `System.Array[2] myArray;`

4. En C#, les `string` (chaînes de caractères) sont...

*1 seule réponse possible*

- ☐ ... imbitables.
- ☐ ... mutables.
- ☒ ... immutables.
- ☐ ... inclassables.

# QCM développement en C#

5. Lorsqu'on applique le Domain Driven Design, à quoi sert la base de données ?

1 seule réponse possible

- ☐ A contenir la logique métier dans des procédures stockées.
- ☐ A donner des droits d'accès aux données aux utilisateurs.
- ☒ A persister les données.
- ☐ A archiver les vieux produits qui ne sont plus en vente.

6. Quelle classe permettrait de corriger la grossière faute que contient le code suivant ?

```
1. IList<Client> clients = ClientsRepository.GetAll();
2. string commaSeparatedClientsName = string.Empty;
3.
4. foreach (Client client in clients)
5. {
6.     commaSeparatedClientsName += client.Name + ",";
7. }
```

1 seule réponse possible

- ☐ ArrayList
- ☒ StringBuilder
- ☐ TagBuilder
- ☐ CSVBuilder

7. Dans un projet contenant une couche métier, une couche d'accès aux données, et une couche applicative, qui référence logiquement quoi ?

1 seule réponse possible

☐ La couche métier référence la couche applicative.  
La couche d'accès aux données référence la couche applicative.  
La couche applicative ne référence rien.

☐ La couche métier référence la couche d'accès aux données.  
La couche d'accès aux données référence la couche métier.  
La couche applicative ne référence rien.

☐ La couche métier référence la couche d'accès aux données.  
La couche d'accès aux données référence la couche applicative.  
La couche applicative référence la couche d'accès aux données.

☒ La couche métier ne référence rien.  
La couche d'accès aux données référence la couche métier.  
La couche applicative référence la couche métier et la couche d'accès aux données.

# QCM développement en C#

## 8. Quelle est la différence entre les delegates Action et Func ?

*1 seule réponse possible*

- ☒ Action n'a pas de type de retour, alors que Func en a un.
- ☐ Action s'exécute instantanément alors que Func s'exécute au bout d'un certain délai défini par le développeur.
- ☐ Action est Threadsafe alors que Func ne l'est pas.
- ☐ Action concerne les méthodes statiques alors que Func concerne les méthodes d'instance.

## 9. Lorsqu'on utilise un ORM (comme Entity Framework ou NHibernate), qu'appelle-t-on le problème du "select N+1" ?

*1 seule réponse possible*

- ☐ C'est lorsque qu'au cours du calcul d'une somme d'agrégat on compte la ligne d'en-tête.
- ☒ C'est lorsque l'on récupère auprès de la source de données une collection d'objets en les requêtant un par un.
- ☐ C'est lorsque l'on récupère la longueur d'une chaîne de caractères en oubliant de tronquer le caractère de terminaison.
- ☐ C'est lorsque l'on utilise une sous-requête alors qu'on pourrait s'en passer.

## 10. Quelle contrainte faudrait-il appliquer sur la méthode GetBillsCount ?

```
1. public interface IBill
2. {
3.     IList<Bill> { get; set; };
4. }
5.
6. public class Client : IBill
7. {
8.     public IList<Bill> Bills { get; set; }
9. }
10.
11. public class Supplier : IBill
12. {
13.     public IList<Bill> Bills { get; set; }
14. }
15.
16. public static class ToolBox
17. {
18.     public static GetBillsCount<T>(T source)
19.     {
20.         return source.Bills.Count;
21.     }
22. }
```

# QCM développement en C#

1 seule réponse possible

- ☐ where T : static
- ☐ where T : interface
- ☐ where T : class
- ☒ where T : IBill

11. Quelle sera la valeur de la variable foobar à la fin de l'exécution de ce code ?

```
1. int? foobar = 42, dummy = null;
2. string text = null;
3. foobar = text != null ? text.Length : 10;
4. foobar = dummy ?? null;
```

1 seule réponse possible

- ☐ 10.
- ☒ null.
- ☐ 4.
- ☐ 42.

12. En Domain Driven Design, que désigne-t-on sous le nom de "domain" ?

1 seule réponse possible

- ☐ L'ensemble des noms de réseaux et sous-réseaux appartenant à l'entreprise.
- ☐ La spécialité à laquelle appartient le chef de projet.
- ☒ L'ensemble des connaissances qui constituent la logique métier.
- ☐ L'ensemble des technologies sélectionnées pour réaliser un projet.

13. Comment appelle-t-on le code suivant ?

```
1. x => x.Length == 12
```

1 seule réponse possible

- ☐ Expression de demie-vie.
- ☐ Expression lambda.
- ☐ Expression adhoc.
- ☒ Query expression.

# QCM développement en C#

14. Lequel de ces types n'est pas un type valeur ?

1 seule réponse possible

- ☐ DateTime
- ☒ string
- ☐ int
- ☐ char

15. Quel est le problème du code suivant ?

```
1. IList<Couple> couples = new List<Couple>();
2.
3. // On utilise Entity Framework pour interroger une base de données
4. using (var entities = new PeopleDBEntities())
5. {
6.     IQueryable<Person> people = from p in entities.People
7.                                 select p;
8.
9.     foreach (Person p1 in people)
10.    {
11.        foreach (Person p2 in people)
12.        {
13.            if (p1.ConjointId = p2.Id)
14.            {
15.                Couple couple = new Couple(p1, p2);
16.
17.                // La classe Couple compare ses instances par rapport
18.                // aux Id des deux personnes qu'elle contient
19.                if (!couples.Contains(couple))
20.                    couples.Add(couple);
21.            }
22.        }
23.    }
24. }
```

1 seule réponse possible

- ☒ Chaque foreach provoquera un appel vers la base de données pour récupérer les mêmes données.
- ☐ On ne tient pas compte des ménages à trois.
- ☐ On fait un produit cartésien des personnes, et ils formeront tous des couples les uns avec les autres.
- ☐ On risque un dépassement de capacité sur la collection couples.

# QCM développement en C#

16. Quel sera le résultat de cette opération ?

```
1. var result = 2 / 3;
```

1 seule réponse possible

- ☐ result est de type double et sa valeur est 0.6666666666666666.
- ☐ result est de type decimal et sa valeur est 0.66666666.
- ☐ result est de type float et sa valeur est 0.6666.
- ☒ result est de type int et sa valeur est 0.

17. Quelle sera la valeur de la variable result ?

```
1. [Flags]
2. public enum RGBColor : int
3. {
4.     Black = 0,
5.     Red = 1,
6.     Green = 2,
7.     Blue = 4
8. }
9.
10. int result = (int)(RGBColor.Red | RGBColor.Blue);
```

1 seule réponse possible

- ☐ 0
- ☐ 1
- ☐ 4
- ☐ 5

18. A partir du Framework .NET 3.5, que permet d'abstraire une interface ?

1 seule réponse possible

- ☒ Les propriétés, les méthodes, les indexeurs et les événements.
- ☐ Les propriétés, les méthodes et les délégués.
- ☐ Les propriétés, les méthodes, les délégués, les énumérations et les événements.
- ☐ Les propriétés et les méthodes.

# QCM développement en C#

19. Quelle est la différence entre ces deux lignes de code ?

```
1. // Cas 1
2. Person person = (Person)something;
3.
4. // Cas 2
5. Person person = something as Person;
```

1 seule réponse possible

- ☐ Si something est bien de type Person, le cas 1 conservera aussi ses propriétés provenant d'autres types, alors que le cas 2 les effacera.
- ☐ Si something n'est pas de type Person, le cas 1 lèvera une exception alors que le cas 2 renverra une nouvelle instance de la classe Person
- ☒ Si something n'est pas de type Person, le cas 1 lèvera une exception alors que le cas 2 renverra null.
- ☐ Il n'y a aucune différence. On effectue juste la même opération de deux manières différentes.

20. Ce code compilé en C# 4.0 va lever une RuntimeBinderException. Mais à quel moment ?

```
1. dynamic title = "My favorite mistake";
2. int numberOfLetters = title.Size;
```

1 seule réponse possible

- ☐ Au moment de saisir le code dans Visual Studio.
- ☐ A la compilation du code.
- ☒ A l'exécution du code.
- ☐ A l'installation du programme.

21. Quel type sera donné à la variable result grâce à l'inférence de type ?

```
1. var result = (int i1, int i2) => (i1 + i2).ToString();
```

1 seule réponse possible

- ☐ Func<int, int, string>.
- ☒ On ne peut pas inférer le type d'une expression lambda.
- ☐ Expression<int, int, string>.
- ☐ Expression<Func<int, int, string>>.

# QCM développement en C#

22. Selon les bonnes pratiques édifiées par Microsoft, lequel de ces délégués peut servir à la création d'un événement ?

```
1. // Délégué 1
2. public delegate void TrainArrivalEventHandler(object sender, EventArgs e);
3.
4. // Délégué 2
5. public delegate bool CallHeadquartersEventHandler(string phoneNumber);
6.
7. // Délégué 3
8. public delegate bool RequestEventHandler(object parameter1, object parameter2,
9.   object parameter3);
10.
11. // Délégué 4
12. public delegate void OperationExecutedEventHandler();
```

1 seule réponse possible

- ☒ Le délégué 1.
- ☐ Le délégué 2.
- ☐ Le délégué 3.
- ☐ Le délégué 4.

23. A quel élément du pattern architectural Domain Driven Design correspond la définition suivante ?

*"Élément immutable sans identité dont on ne souhaite pas conserver l'historique".*

1 seule réponse possible

- ☒ Value object.
- ☐ Entity.
- ☐ Aggregate.
- ☐ Repository.

24. Si une méthode est marquée comme protected internal, qui peut y accéder ?

1 seule réponse possible

- ☐ Les classes qui sont à la fois dans le même assembly et qui dérivent de la classe dans laquelle cette méthode est déclarée.
- ☐ Seules les méthodes qui sont dans la même classe que la méthode en question.
- ☐ Les méthodes Internal ne peuvent être appelées que par réflexion.
- ☒ Les classes du même assembly et celles qui dérivent de la classe dans laquelle cette méthode est déclarée.



# QCM développement en C#

---

## 25. Qu'est-ce que le boxing ?

*1 seule réponse possible*

- ☐ Encapsuler un objet dans un type valeur
- ☐ Encapsuler une copie d'un objet dans un type valeur
- ☐ Encapsuler un type valeur dans un objet
- ☒ Encapsuler la copie d'un type valeur dans un objet

## 26. Quel paramètre du compilateur permet de créer un fichier xml à partir des commentaires xml des fichiers d'un assembly ?

*1 seule réponse possible*

- ☐ /text
- ☒ /doc
- ☐ /xml
- ☐ /help

## 27. Qu'est-ce qu'un assembly satellite ?

*1 seule réponse possible*

- ☐ Un assembly périphérique désigné pour monitorer les requêtes de permission pour une application.
- ☐ N'importe quelle DLL utilisée par un fichier EXE.
- ☒ Un assembly contenant des ressources localisées pour un autre assembly.
- ☐ Un assembly désigné pour modifier l'apparence ou la skin d'une application.

## 28. Qu'est-ce qu'un délégué ?

*1 seule réponse possible*

- ☒ Un pointeur de fonction fortement typé.
- ☐ Un thread ou un process qui peut appeler une seule méthode.
- ☐ Une référence à un objet dans un process différent.
- ☐ Un message inter-process.

# QCM développement en C#

29. Comment le versionnement des assemblies permet d'éviter l'enfer des DLL ?

1 seule réponse possible

- ☐ L'environnement d'exécution vérifie qu'une seule version de l'assembly est présente à la fois sur une machine.
- ☒ Le .NET permet aux assemblies de spécifier le nom et la version des assemblies qu'ils ont besoins d'utiliser.
- ☐ Le compilateur vérifie la compatibilité descendante lors la compilation.
- ☐ Il ne le permet pas.

30. Quel patron de conception est présenté ci-dessous ?

```
1. public class A {  
2.     private A instance;  
3.     private A() {  
4.     }  
5.     public static A Instance {  
6.         get  
7.         {  
8.             if (instance == null)  
9.                 instance = new A();  
10.            return instance;  
11.        }  
12.    }  
13. }
```

1 seule réponse possible

- ☐ Factory
- ☐ Abstract Factory
- ☒ Singleton
- ☐ Builder

31. Dans la framework de test NUnit, quel attribut doit décorer une classe de test pour quelle soit présente dans l'interface graphique de NUnit ?

1 seule réponse possible

- ☐ TestAttribute
- ☐ TestClassAttribute
- ☒ TestFixtureAttribute
- ☐ NUnitTestClassAttribute

# QCM développement en C#

---

**32. Laquelle des opérations suivantes ne peut-on pas faire avec un DataSet ADO .NET ?**

*1 seule réponse possible*

- ☐ Synchroniser le DataSet avec une base de données
- ☐ Synchroniser le DataSet avec un RecordSet
- ☐ Convertir le DataSet en XML
- ☒ Déterminer le schéma de données d'un DataSet

**33. En programmation orientée objet, comment décririez-vous l'encapsulation ?**

*1 seule réponse possible*

- ☐ La conversion d'un type d'objet vers un autre.
- ☐ La résolution à la volée d'appels de méthode.
- ☐ L'exposition des données.
- ☒ La séparation de l'interface de l'implémentation