SKILLS CAMPUS
Training & Certification Center

Optimisation d'applications en .NET

-Réference: **SII-311** -Durée: **4 Jours (28 Heures)**

Les objectifs de la formation

- Maîtriser l'architecture des applications
- Net
-
- Diagnostiquer les allocations mémoire et éviter les fuites mémoire
-
- Optimiser le code C# en référence aux dernières versions C# 7, 8 et 9
-
- Maîtriser les opérations sur les tableaux et les collections
-
- Optimiser et mesurer les gains de performances via le parallélisme
-
- Coder en C# pour résoudre des problématiques réelles
-

A qui s'adesse cette formation ?

POUR QUI :

- Développeurs, ingénieurs, architectes, chefs de projet.

PRÉREQUIS :

- Bonnes connaissances du langage C#.
- Expérience requise.

Programme

- **Introduction**
  - Évolution de la plateforme .Net et de son écosystème.
  - Optimiser : quoi, comment, quand ? Objectifs.
- **Le ramasse-miettes GC, optimisation des allocations et diagnostic**
  - La gestion de la mémoire par le ramasse-miettes et API GC.
  - Outils de diagnostic de Visual Studio, de JetBrains et BenchmarkDotNet.

Programme

- Le Pattern Dispose et l'implémentation de IDisposable.
- Recommandations pour éviter les fuites mémoires potentielles d'une application .Net.
- Garde-fous au design time via les Règles d'Analyse de code.
- Outils de l'écosystème .Net pour tracer les erreurs de boxing, de closure et références nulles.

- **Méthodes C# et constructions du langage**
  - Méthodes C# et l'évolution du codage.
  - Méthodes d'extension et architecture LINQ.
  - Méthodes asynchrones et utilisation des mots clés async/await.
  - Mesures de performance des allocations des types Value comparés aux types Reference.
  - Nouveau type valeur ValueTuple de C# 7.0.
  - Passage de paramètres et retour par référence de C# 7.2.
  - Avantages de la programmation fonctionnelle en C# et comparaison au langage fonctionnel F#.
  - Pattern Matching en C#.

- **Tableaux et collections**
  - Performance des opérations sur les données.
  - Collections du .Net et caractéristiques.
  - Implémentation de l'interface IEquatable.
  - Pointeur IntPtr et le mot clé C# 7.2 stackalloc pour allouer sur la pile.

- **La librairie TPL du .Net 4.7 : asynchronisme et parallélisme**
  - Évolution de la gestion des appels asynchrones via les mots clés async/await.
  - Les nouvelles classes de System.Threading.Tasks.
  - Paralléliser les itérations for et foreach. Design pattern en matière de parallélisme.
  - Utilisation de l'infrastructure PLinq.
  - Diagnostic de performance via le NuGet BenchmarkDotNet et DotTrace de JetBrains.

- **Conclusion**
  - Incidence des fonctionnalités C# 8 et C# 9 sur le code et les applications .Net.
  - Une synthèse et quelques conseils.
  - Recommandations d'ouvrages et références internet.

(+212) 5 22 27 99 01

(+212) 6 60 10 42 56

Contact@skills-group.com

Nous sommes à votre disposition :
De Lun - Ven 09h00-18h00 et Sam 09H00 – 13H00

Angle bd Abdelmoumen et rue Soumaya, Résidence Shehrazade 3, 7éme étage N° 30
Casablanca 20340, Maroc