

Optimization of applications in .NET

SII-311 4 Days (28 Hours)



Description

This training presents a methodology to improve the efficiency of your .NET and .Net Core applications. You will learn how to use the most important performance analysis and diagnostic tools, and you will master the different code optimization techniques in C# language. This will allow you to optimize your applications so that they work faster and more efficiently.

Who is this training for ?

For whom

Developers, engineers, architects, project managers.

Prerequisites

Good knowledge of the C# language. Experience required.

Training objectives

- Master the architecture of
- Net applications
- · Diagnose memory allocations and avoid memory leaks
- Optimize C# code with reference to the latest C# versions 7, 8 and 9
- · Master operations on arrays and collections
- Optimize and measure performance gains via parallelism
- Code in C# to solve real-world problems

Training program

Introduction

- Evolution of the .Net platform and its ecosystem.
- Optimize: what, how, when? Objectives.

lana(de2112):5022n27099y0/e unknown

Whatsappt fo(+212) 6/60 10.4256n

maile Contact@skillsrgroup.comn

image norther of bot Abdelmoumen and rue Soumaya, Shehrazade 3 Residence, 7th floor Nº 30, Casablanca 20340, Morocco



Le ramasse-miettes GC, optimisation des allocations et diagnostic

- Memory management by garbage collection and GC API.
- Diagnostic tools from Visual Studio, JetBrains and BenchmarkDotNet.
- The Pattern Disposes and the implementation of IDisposable.
- Recommendations to avoid potential memory leaks in a .Net application.
- Guards to design time via Code Analysis Rules.
- Tools from the .Net ecosystem to trace boxing, closure and null reference errors.

Méthodes C# et constructions du langage

- C# methods and the evolution of coding.
- Extension methods and LINQ architecture.
- Asynchronous methods and the use of async/await keywords.
- Performance metrics for Value type allocations compared to Reference types.
- New ValueTuple value type in C# 7.0.
- Parameter passing and returning by reference in C# 7.2.
- Advantages of functional programming in C# and comparison to the functional language F#.
- Pattern Matching in C#.

Tableaux et collections

- Performance of data operations.
- .Net collections and characteristics.
- Implementation of the IEquatable interface.
- IntPtr pointer and the C# 7.2 stackalloc keyword to allocate on the stack.

La librairie TPL du .Net 4.7 : asynchronisme et parallélisme

- Evolution of the management of asynchronous calls via the async/await keywords.
- The new classes of System.Threading.Tasks.
- Parallelize the iterations for and foreach. Design pattern in terms of parallelism.
- Using the PLinq infrastructure.
- Performance diagnosis via JetBrains BenchmarkDotNet and DotTrace NuGet.

Conclusion

- Impact of C# 8 and C# 9 features on .Net code and applications.
- A summary and some advice.
- · Book recommendations and internet references .

Emaile nGotner of bdt Abdelmoumen and rue Soumaya, Shehrazade 3 Residence, 7th floor Nº 30, Casablanca 20340, Morocco