# Вопросы для самопроверки

Список вопросов для самопроверки по .NET. За основу взята статья Скота Ханселмана [What Great .NET Developers Ought To Know (More .NET Interview Questions)](http://www.hanselman.com/blog/WhatGreatNETDevelopersOughtToKnowMoreNETInterviewQuestions.aspx).

## Общие вопросы

* В чем разница между процессом (process) и потоком (thread)?
* Что такое сервис windows (windows service) и чем он отличается от обычного исполняемого файла?
* В чем заключается разница между EXE и DLL файлами?
* Что такое GAC? Для решения каких задач он был создан?
* Объясните назначение ключевых слов virtual, sealed, override и abstract.
* В чем различие между catch(Exception e){throw e;} и catch(Exception e){throw;}?
* Опишите особенности класса string. Есть ли разница между string и System.String?
* Для чего используется класс StringBuilder?

## Средний уровень

* Что такое интерфейс и в чем его отличие от класса?
* В чем заключается разница между интерфейсом и абстрактным классом?
* Что такое отражение (reflection)?
* Может ли переменная типа DateTime принимать значение null?
* Что вам известно о JIT и NGEN? Что они делают? Опишите их плюсы и минусы.
* Каким образом сборщик мусора (garbage collector) управляет жизненным периодом объектов в .NET? Что означает термин недетерминированное освобождение (non-deterministic finalization)?
* В чем польза using() { … }? Для чего используется IDisposable? Как он помогает выполнить детерминированное освобождение (deterministic finalization)? Напишите пример реализации класса-наследника IDisposable.
* Чем отличаются Finalize() и Dispose()?

**Everyone who writes code**

* ~~Describe the difference between a Thread and a Process?~~
* ~~What is a Windows Service and how does its lifecycle differ from a "standard" EXE?~~
* What is the maximum amount of memory any single process on Windows can address? Is this different than the maximum virtual memory for the system? How would this affect a system design?
* What is the difference between an EXE and a DLL?
* What is strong-typing versus weak-typing? Which is preferred? Why?
* Corillian's product is a "Component Container." Name at least 3 component containers that ship now with the Windows Server Family.
* What is a PID? How is it useful when troubleshooting a system?
* How many processes can listen on a single TCP/IP port?
* What is the GAC? What problem does it solve?

**Mid-Level .NET Developer**

* Describe the difference between Interface-oriented, Object-oriented and Aspect-oriented programming.
* ~~Describe what an Interface is and how it’s different from a Class.~~
* ~~What is Reflection?~~
* What is the difference between XML Web Services using ASMX and .NET Remoting using SOAP?
* Are the type system represented by XmlSchema and the CLS isomorphic?
* Conceptually, what is the difference between early-binding and late-binding?
* Is using Assembly.Load a static reference or dynamic reference?
* When would using Assembly.LoadFrom or Assembly.LoadFile be appropriate?
* What is an Asssembly Qualified Name? Is it a filename? How is it different?
* Is this valid? Assembly.Load("foo.dll");
* How is a strongly-named assembly different from one that isn’t strongly-named?
* ~~Can DateTimes be null?~~
* ~~What is the JIT? What is NGEN? What are limitations and benefits of each?~~
* ~~How does the generational garbage collector in the .NET CLR manage object lifetime? What is non-deterministic finalization?~~
* ~~What is the difference between Finalize() and Dispose()?~~
* How is the using() pattern useful? What is IDisposable? How does it support deterministic finalization?
* What does this useful command line do? tasklist /m "mscor\*"
* What is the difference between in-proc and out-of-proc?
* What technology enables out-of-proc communication in .NET?
* When you’re running a component within ASP.NET, what process is it running within on Windows XP? Windows 2000? Windows 2003?

**Senior Developers/Architects**

* What’s wrong with a line like this? DateTime.Parse(myString);
* What are PDBs? Where must they be located for debugging to work?
* What is cyclomatic complexity and why is it important?
* Write a standard lock() plus “double check” to create a critical section around a variable access.
* What is FullTrust? Do GAC’ed assemblies have FullTrust?
* What benefit does your code receive if you decorate it with attributes demanding specific Security permissions?
* What does this do? gacutil /l | find /i "Corillian"
* What does this do? sn -t foo.dll
* What ports must be open for DCOM over a firewall? What is the purpose of Port 135?
* Contrast OOP and SOA. What are tenets of each?
* How does the XmlSerializer work? What ACL permissions does a process using it require?
* Why is catch(Exception) almost always a bad idea?
* What is the difference between Debug.Write and Trace.Write? When should each be used?
* What is the difference between a Debug and Release build? Is there a significant speed difference? Why or why not?
* Does JITting occur per-assembly or per-method? How does this affect the working set?
* Contrast the use of an abstract base class against an interface?
* What is the difference between a.Equals(b) and a == b?
* In the context of a comparison, what is object identity versus object equivalence?
* How would one do a deep copy in .NET?
* Explain current thinking around IClonable.
* What is boxing?
* Is string a value type or a reference type?
* What is the significance of the "PropertySpecified" pattern used by the XmlSerializer? What problem does it attempt to solve?
* Why are out parameters a bad idea in .NET? Are they?
* Can attributes be placed on specific parameters to a method? Why is this useful?

**C# Component Developers**

* Juxtapose the use of override with new. What is shadowing?
* Explain the use of virtual, sealed, override, and abstract.
* Explain the importance and use of each component of this string: Foo.Bar, Version=2.0.205.0, Culture=neutral, PublicKeyToken=593777ae2d274679d
* Explain the differences between public, protected, private and internal.
* What benefit do you get from using a Primary Interop Assembly (PIA)?
* By what mechanism does NUnit know what methods to test?
* What is the difference between: catch(Exception e){throw e;} and catch(Exception e){throw;}
* What is the difference between typeof(foo) and myFoo.GetType()?
* Explain what’s happening in the first constructor: public class c{ public c(string a) : this() {;}; public c() {;} } How is this construct useful?
* What is this? Can this be used within a static method?

**ASP.NET (UI) Developers**

* Describe how a browser-based Form POST becomes a Server-Side event like Button1\_OnClick.
* What is a PostBack?
* What is ViewState? How is it encoded? Is it encrypted? Who uses ViewState?
* What is the <machinekey> element and what two ASP.NET technologies is it used for?
* What three Session State providers are available in ASP.NET 1.1? What are the pros and cons of each?
* What is Web Gardening? How would using it affect a design?
* Given one ASP.NET application, how many application objects does it have on a single proc box? A dual? A dual with Web Gardening enabled? How would this affect a design?
* Are threads reused in ASP.NET between reqeusts? Does every HttpRequest get its own thread? Should you use Thread Local storage with ASP.NET?
* Is the [ThreadStatic] attribute useful in ASP.NET? Are there side effects? Good or bad?
* Give an example of how using an HttpHandler could simplify an existing design that serves Check Images from an .aspx page.
* What kinds of events can an HttpModule subscribe to? What influence can they have on an implementation? What can be done without recompiling the ASP.NET Application?
* Describe ways to present an arbitrary endpoint (URL) and route requests to that endpoint to ASP.NET.
* Explain how cookies work. Give an example of Cookie abuse.
* Explain the importance of HttpRequest.ValidateInput()?
* What kind of data is passed via HTTP Headers?
* Juxtapose the HTTP verbs GET and POST. What is HEAD?
* Name and describe at least a half dozen HTTP Status Codes and what they express to the requesting client.
* How does if-not-modified-since work? How can it be programmatically implemented with ASP.NET?  
  Explain <@OutputCache%> and the usage of VaryByParam, VaryByHeader.
* How does VaryByCustom work?
* How would one implement ASP.NET HTML output caching, caching outgoing versions of pages generated via all values of q= except where q=5 (as in <http://localhost/page.aspx?q=5>)?

**Developers using XML**

* What is the purpose of XML Namespaces?
* When is the DOM appropriate for use? When is it not? Are there size limitations?
* What is the WS-I Basic Profile and why is it important?
* Write a small XML document that uses a default namespace and a qualified (prefixed) namespace. Include elements from both namespace.
* What is the one fundamental difference between Elements and Attributes?
* What is the difference between Well-Formed XML and Valid XML?
* How would you validate XML using .NET?
* Why is this almost always a bad idea? When is it a good idea? myXmlDocument.SelectNodes("//mynode");
* Describe the difference between pull-style parsers (XmlReader) and eventing-readers (Sax)
* What is the difference between XPathDocument and XmlDocument? Describe situations where one should be used over the other.
* What is the difference between an XML "Fragment" and an XML "Document."
* What does it meant to say “the canonical” form of XML?
* Why is the XML InfoSet specification different from the Xml DOM? What does the InfoSet attempt to solve?
* Contrast DTDs versus XSDs. What are their similarities and differences? Which is preferred and why?
* Does System.Xml support DTDs? How?
* Can any XML Schema be represented as an object graph? Vice versa?