1. **Что такое .Net Framework и из чего он состоит?**

Платформа .NET Framework —программная. Платформа .на с# .NET Framework состоит из общеязыковой среды выполнения (среды CLR),ASP и библиотеки классов .NET Framework.

1. **Поясните, что такое CLR-среда. Что такое FCL? Что такое IL?**

CLR (Common Language Runtime) – общеязыковая среда исполнения, виртуальная машина на которой исполняются все приложения, работающие в среде .NET.

FCL (.NET Framework Class Library) – соответствующая CLS спецификации объектно-ориентированная библиотека классов, интерфейсов и системы типов (типов-значений)

Сборка-дискретная единица часто используемого кода

IL - объектно-ориентированный машинный язык не зависящий от процессора

1. **Пояснить работу JIT-компилятора?**

Здесь и сейчас,компиляция на летуИнтерпретирует IL код в машинные команды

1. **Что такое CTS (Common Type System)**

Common Type System (сокр. CTS, [рус.](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D1%81%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Общая система типов) — часть [.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework), формальная спецификация, определяющая, как какой-либо тип (класс, интерфейс, структура, встроенный тип данных) должен быть определён для его правильного выполнения средой .NET.

1. **Какие аспекты поведения определяет тип System.Object**

Поддерживает все классы в иерархии классов .NET Framework и предоставляет низкоуровневые службы для производных классов. Он является исходным базовым классом для всех классов платформы .NET Framework и корнем иерархии типов.

1. **Что находится в MSCorLib dll?**

mscorlib определенно является одной из библиотек базового класса .net, и каждая программа на С# зависит от нее

1. **Что такое сбока? Из чего состоит сборка .NET?**

**Сборка** является базовой структурной единицей в .NET, на уровне которой проходит контроль версий, развертывание и конфигурация приложения.

Сборки кристаллизуют всю библиотеку классов .NET - при написании кода и создании сборки своего приложения мы используем пространства имен, которые размещены в других сборках .NET.

Сборки имеют следующие составляющие:

* Манифест, который содержит метаданные сборки
* Метаданные типов. Используя эти метаданные, сборка определяет местоположение типов в файле приложения, а также места размещения их в памяти
* Собственно код приложения на языке MSIL, в который компилируется код C#
* Ресурсы

1. Какие виды сборок существуют?

Приватные сборки

## Разделяемые сборки

## Однофайловые и многофайловые сборки

1. Что такое assembly manifest?

**Манифест сборки** (assembly manifest) - это внутренняя часть сборки, которая позволяет ей быть самоописанной. Assembly manifest позволяет идентифицировать сборку, указывает файлы, которые включаются в реализацию сборки, описывает типы и ресурсы, используемые в сборке, указывает зависимости от других сборок, а также набор прав доступа, которые необходимы сборке для корректной работы. Эта информация используется во время выполнения для разрешения ссылок, проверку корректности версий, проверку целостности загруженных сборок.

1. Что такое GAC?

Global Assembly Cache (англ. GAC — Глобальный Кэш Сборок) — в инфраструктуре .NET — подсистема, хранящая сборки CLI (CLI assembly) в централизованном репозитории.

1. Чем managed code отличается от unmanaged code

Управляемый код выполняется под управлением [CLR](http://alexeyworld.com/blog/functioning_of_the_clr.109.aspx)-системы. **Неуправляемый код - это обычный машинный код**.

1. Как и для чего определен метод Main?

Метод с именем Main является самым **главным методом** в программе на языке программирования c sharp (c#), потому что с него начинается выполнение приложения.

1. Варианты использования директивы using( using Directive ) в C#.

Директива **using** используется в двух случаях:

разрешает использование типов в пространстве имен, поэтому уточнение использования типа в этом пространстве имен не требуется; позволяет создавать псевдонимы пространства имен или типа. Это называется *директива using alias*.

1. Как связаны между собой сборки и пространства имен?

Namespace (пространство имен) является логическим соглашением, используемым во время разработки, в то время как assembly (сборка) устанавливает область видимости имени в процессе выполнения.

1. Что такое примитивные типы данных? Перечислите их.

Типы данных, которые поддерживаются компилятором напрямую, называются примитивными.

Bool, byte, int, float, double, decimal, char.

1. Что такое ссылочные типы? Какие типы относятся к ним?

Ссылочные типы хранят ссылки на их объекты.

Object, string, class.

1. Какие типы относятся к типам-значениям?

Целочисленные типы, типы с плавающей точкой, bool, char, struct.

1. В чем отличие между ссылочными и значимыми типами данных?

Ссылочные типы хранят ссылки на их объекты, а значимые типы содержат свои данные непосредственно.

1. Что такое упаковка и распаковка значимых типов?

Упаковка и распаковка значимых типов в C# — доступный для программиста механизм преобразования размерных типов данных языка C# из значимых в ссылочные и обратно

1. В чем заключается разница между int и System.Int32? double и System.Double и т.д.?

Разницы нет.

1. Для чего используется тип dynamic?

Отражения или коммуникации с другими компонентами. Может получить какое угодно начальное значение, и на протяжении времени его существования это значение может быть заменено новым.

1. В чем заключается главное отличие между var и dynamic?

В **C# 4.0**вводится новый тип, известный как **динамический тип**. Он используется для того, чтобы избежать проверки типа во время компиляции. Компилятор не проверяет тип переменной динамического типа во время компиляции, вместо этого компилятор получает тип во время выполнения. Переменная динамического типа создается с помощью ключевого слова dynamic.

1. Что такое неявно типизированная переменная?

Переменная типа var. Компилятор сам выбирает наиболее подходящий тип.

1. Для чего используют Nullable тип?

Для присвоения типам значений null

1. Как объявить строковый литерал? Какие операции можно выполнять со строкой?

Задается ковычках…Склеивание, сравнение строк, поиск в строке, разделение строк, замена, удаление строк, смена регистра.

1. Какие есть способы для задания и инициализации строк?

С помощью строкового литерала и StringBuilder.

1. Какие методы есть у типа String?

Compare, concat, insert, replace, split, и т.д.

1. В чем отличие пустой и null строки?

null - переменная не определена, '' - переменная содержит пустое значение.

1. Как можно выполнить сравнение строк?

С помощью метода compare или оператора ==

1. В чем отличие типов String и StringBuilder?

String нельзя модифицировать, но можно заменить созданием нового экземпляра.

StringBuilder модифицироавть можно.

1. . Поясните явные преобразования переменных с помощью команд Convert.

 Класс Convert представляет еще один способ для преобразования значений. Для этого в нем определены следующие статические методы. Если методу не удастся преобразовать значение к нужному типу, то он выбрасывает исключение FormatException.

1. Как выполнить консольный ввод/вывод?

Console.WriteLine ; Console.ReadLine.

1. Приведите примеры определения и инициализации одномерных и двумерных массивов

int[] nums2 = new int[4] { 1, 2, 3, 5 };

int[,] mas = { { 1, 2, 3 }, { 4, 5, 6 }, { 7, 8, 9 }, { 10, 11, 12 } };

1. Что такое ступенчатый массив? Как его задать?

Ступеньчатый массив – массив массивов. Задается с помощью парных квадратных скобок new int[3][]. массивы. Причем длина массива указывается только в первых квадратных скобках, все последующие квадратные скобки должны быть пусты.

1. Какие типы можно использовать в foreach? Приведите пример.

Массивы, коллекции.StringBuilder объекты,String и string переменные

**foreach** (**var** item **in** sb1)

            {

                Console.WriteLine(item);

            }

1. Что такое кортеж? Для чего и как он используется?

Кортеж – набор данных разного типа. Часто используется при необходимости возвращения из функции нескольких значений.

1. Что такое локальная функция?

Локальные функции представляют функции, определенные внутри других методов.

38. В чем разница между кодом, заключенным в блок checked и кодом, заключенным в блок unchecked?

В C# допускается указывать, будет ли в коде сгенерировано исключение при переполнении, с помощью ключевых слов checked и unchecked. Так, если требуется указать, что выражение будет проверяться на переполнение, следует использовать **ключевое слово checked**, а если требуется проигнорировать переполнение — **ключевое слово unchecked**. В последнем случае результат усекается, чтобы не выйти за пределы диапазона представления чисел для целевого типа выражения.

39. Какой контекст (checked/unchecked) применяется по умолчанию? Как можно переопределить это поведение?

Если ни checked, ни unchecked не указаны, контекст по умолчанию для неконстантных выражений (выражения, вычисляемые во время выполнения) определяется по значению параметра компилятора [-checked](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/compiler-options/checked-compiler-option). По умолчанию значение этого параметра не задано, и арифметические операции выполняются в непроверенном контексте.

Для константных выражений (выражения, которые можно полностью вычислить во время компиляции) контекстом по умолчанию является проверяемый. Если только константное выражение явным образом не размещено в непроверенном контексте, переполнения, возникающие при вычислении выражения во время компиляции, вызывают ошибки времени компиляции.

40. Для чего используется ключевое слово fixed? Каковы особенности его использования?

Оператор fixed задает указатель на управляемую переменную и "закрепляет" эту переменную во время выполнения оператора. Указатели на перемещаемые управляемые переменные полезны только в контексте fixed. Без контекста fixed при сборке мусора эти переменные могут переноситься непредсказуемым образом. Компилятор C# позволяет присвоить указатель только управляемой переменной в операторе fixed.