**Вопросы**

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

Программная платформа, позволяющая программисту не задумываться об особенностях системы, на которой будет запускаться программа. Основой является исполняющая среда Common Language Runtime (CLR), Вторая, не менее важная часть, это библиотека классов Framework Class Library (FCL).

1. **Что такое CLR, FCL/BCL, CLI, IL?**

Среда **CLR** управляет памятью, выполнением потоков, выполнением кода, проверкой безопасности кода, компиляцией и другими системными службами. Эти средства являются внутренними для управляемого кода, который выполняется в среде CLR.

**Base Class Library**, или **FCL** (англ. Framework Class Library), — стандартная библиотека классов платформы «[.NET Framework](https://ru.wikipedia.org/wiki/.NET_Framework)». Программы, написанные на любом из языков, поддерживающих платформу .NET, могут пользоваться классами и методами BCL — создавать объекты классов, вызывать их методы, наследовать необходимые классы BCL и т. д.

**CLI** ( *Common Language Infrastructure*) — спецификация общеязыковой инфраструктуры, определяющая архитектуру исполнительной системы .NET — [CLR](https://ru.wikipedia.org/wiki/CLR) и сервисы, предоставляемые CLR выполняемым программам, классы, предоставляемые библиотекой [BCL](https://ru.wikipedia.org/wiki/Base_Class_Library), синтаксис и мнемонику общего промежуточного языка ([CIL](https://ru.wikipedia.org/wiki/MSIL)).

**Common Intermediate Language** (сокращённо **CIL**) — «высокоуровневый ассемблер» виртуальной машины [.NET](https://ru.wikipedia.org/wiki/Microsoft_.NET). Промежуточный язык. Все [компиляторы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%B8%D0%BB%D1%8F%D1%82%D0%BE%D1%80), поддерживающие платформу [.NET](https://ru.wikipedia.org/wiki/Microsoft_.NET), должны [транслировать](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%B0%D0%BD%D1%81%D0%BB%D1%8F%D1%82%D0%BE%D1%80) код с [языков высокого уровня платформы .NET](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_.NET-%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2) на язык CIL.

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

JIT-компиляция ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Just-in-time compilation, компиляция «на лету»), динамическая компиляция ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) dynamic translation) — технология увеличения производительности программных систем, использующих [байт-код](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B9%D1%82-%D0%BA%D0%BE%D0%B4), путём компиляции байт-кода в [машинный код](https://ru.wikipedia.org/wiki/%D0%9C%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BA%D0%BE%D0%B4) или в другой формат непосредственно во время работы программы.

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

часть [.NET Framework](https://dic.academic.ru/dic.nsf/ruwiki/56724), 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.

* Изменение типа
* Проверка на равенство
* Get type
* toString

1. **Что находится в mscorlib dll?**

MsCoRLib - это аббревиатура от **M**icro**s**oft **Co**mmon Language **R**untime **Lib**rary. Mscorlib.dll - это файл DLL в Windows. DLL - это аббревиатура от **D**ynamic **L**ink **L**ibrary (динамически подключаемая библиотека). DLL файлы необходимы программам или расширениям веб-браузера, потому что они содержат программный код, данные и ресурсы.

Вообще, это сборка, в которой содержится большое количество базовых типов, охватывающих широкий спектр типичных задач программирования, а также базовых типов данных, применяемых во всех языках .NET. При построении .NET-решений доступ к этой конкретной сборке будет предоставляться автоматически.

1. **Что такое «сборка»? Из чего состоит сборка .NET?**

**Сборка** — это набор связанных классов, обычно сгруппированных вместе некоторым способом. Сборки можно использовать для отделения частей системы одна от другой. Пространство имен используется для логической группировки родственных типов

Сборки представляют собой базовые элементы развертывания, управления версиями, повторного использования, назначения областей активации и прав доступа для приложений на основе платформы .NET. Сборка представляет собой коллекцию типов и ресурсов, собранных для совместной работы и образующих логическую функциональную единицу. Сборки создаются в виде исполняемого файла (EXE) или файла библиотеки динамической компоновки (DLL) и являются стандартными блоками приложений .NET. Они предоставляют сведения для среды CLR, которые нужны для распознавания реализаций типов.

В .NET Core и .NET Framework сборку можно создать из одного или нескольких файлов исходного кода. В .NET Framework сборки могут содержать один или несколько модулей. Благодаря этому в крупных проектах несколько разработчиков могут работать с отдельными файлами или модулями исходного кода, которые вместе образуют единую сборку.

1. **Какие виды сборок существуют?**

Частные и общие сборки. Частные находятся в каталоге программы, а общие в GAC

1. **Что такое assembly manifest?**

Это внутренняя часть сборки, которая позволяет идентифицировать сборку, указывает файлы, которые включаются в реализацию сборки, описывает типы и ресурсы, используемые в сборке, указывает зависимости от других 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**

В общем случае при написании [С#](http://alexeyworld.com/blog/c_sharp.99.aspx)-программы создается код, называемый **управляемым (managed code).** Управляемый код выполняется под управлением [CLR](http://alexeyworld.com/blog/functioning_of_the_clr.109.aspx)-системы. **Неуправляемый код - это обычный машинный код**. Практически все программы написанные не на базе .NET используют неуправляемый код.

1. **Как и для чего определен метод Main?**

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

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

Директива using разрешает использование типов в пространстве имен, поэтому уточнение использования типа в этом пространстве имен не требуется. Также директива позволяет создать псевдонимы пространства имен или типа. Эта версия использования называется директива с using alias.

1. **Как связаны между собой сборки и пространства имен?**

Сборки и пространства имен не имеют жесткой связи на уровне языка и платформы. Пространства имен - всего лишь приставка к имени типа для явного указания принадлежности типа к определенной группе логически связанных типов.

1. **Что такое примитивные типы данных? Перечислите их.**

Это типы данных, которые поддерживаются компилятором напрямую.byte, short, char, int, long

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

Переменные ссылочного типа содержат в себе ссылки на фактические данные и при этом ссылка указывает на определенную область в памяти, которая была выделена при создании такой переменной.

Ссылочные типы:

* Тип object
* Тип string
* Классы (class)
* Интерфейсы (interface)
* Делегаты (delegate)

1. **Какие типы относятся к типам-значениям?**

- Целочисленные типы (byte, sbyte, char, short, ushort, int, uint, long, ulong)

- Типы с плавающей запятой (float, double)

- Тип decimal

- Тип bool

- Структуры (struct)

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

Значимые типы хранят значение, а ссылочные - ссылку на значение.

1. **Что такое упаковка и распаковка значимых типов?**

Упаковка представляет собой процесс преобразования [типа значения](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/value-types) в тип object или в любой другой тип интерфейса, реализуемый этим типом значения.

Распаковка является явным преобразованием из типа object в [тип значения](https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/value-types) или из типа интерфейса в тип значения, реализующего этот интерфейс.

Позволяет использовать типы-значения в коллекциях(где элементы являются элементами типа object)

Внутренний механизм, который обеспечивает возможность вызывать для типов-значений, подобных int и struct, методы Object

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

Разницы никакой, int и Int32 после компиляции в *IL* код станут одним и тем же. int это ключевое слово для c#, ровно как Integer для *VisualBasic*, но оба они представляют собой System.Int32.

1. **Для чего используется тип dynamic?**

Ключевое слово dynamic позволяет использовать переменные, которые не зависят от типа, а действуют как тип данных, которые они содержат. Динамической переменной можно легко присвоить новое значение, и если тип присваиваемого значения отличается от текущего, динамическая переменная просто автоматически изменит тип. Динамические переменные не проверяются компилятором, что позволяет получить доступ к свойствам, которые могут или не могут присутствовать.

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

Тип var задаётся при инициализации переменной, а dynamic может менять тип в процессе работы.

1. **Что такое неявно типизированная переменная?**

Переменные, объявленные посредствам ключевого слова var.

1. **Для чего используют Nullable тип?**

Значение null по умолчанию могут принимать только объекты ссылочных типов. Однако в различных ситуациях бывает удобно, чтобы объекты числовых типов данных имели значение null, то есть были бы не определены. Стандартный пример - работа с базой данных, которая может содержать значения null. И мы можем заранее не знать, что мы получим из базы данных - какое-то определенное значение или же null. Для этого надо использовать знак вопроса ? после типа значений.

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

string columns = "Column 1\tColumn 2\tColumn 3";

//Output: Column 1 Column 2 Column 3

Строковый литерал – это последовательность символов, заключенная в кавычки.

Обрезка (трим), разделение(сплит), сравнение, копирование, поиск, вставка, удаление, замена, смена регистра, объединение

1. **Какие есть способы для задания и инициализации строк?**

string s1 = "hello”;

string s2 = new String('a', 6); // результатом будет строка "aaaaaa"

string s3 = new String(new char[]{'w', 'o', 'r', 'l', 'd'});

1. **Какие методы есть у типа String?**

Compare: сравнивает две строки

CompareOrdinal: сравнивает две строки без учета локали

Contains: определяет, содержится ли подстрока в строке

Concat: соединяет строки

CopyTo: копирует часть строки или всю строку в другую строку

1. **В чем отличие пустой и null строки?**

null - это отсутствие какого-либо значения. Пустая строка, хоть и пустая, но память под неё в куче выделена, и вы можете к ней обращаться.

1. **Как можно выполнить сравнение строк?**

Compare, Compareordinal.

1. **В чем отличие типов String и StringBuilder?**

StringBuilder – динамическая строка. Класс который находится в пространстве имен System.Text. Емкость увеличивается динамически при изменении строки от 16 до 32 символов с запасом, далее – подстраиваясь под новые размеры строки.

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

Дописывается нужный метод, в который параметром передается значение какого-либо типа, которое нужно конвертировать.

1. **Как выполнить консольный ввод/вывод?**

Console.write/read.

1. **Приведите примеры определения и инициализации одномерных и двумерных массивов.**

int mas[5] = {1, -5, 10, 243, -58};

// Two-dimensional array.

int[,] array2D = new int[,] { { 1, 2 }, { 3, 4 }, { 5, 6 }, { 7, 8 } };

1. **Что такое ступенчатый массив? Как его задать?**

Ступенчатый массив представляет собой массив массивов, в котором длина каждого массива может быть разной.

int[][] myArr = new int[4][];

myArr[0] = new int[4];

myArr[1] = new int[6];

myArr[2] = new int[3];

myArr[3] = new int[4];

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

Массивы или иначе коллекции. Тип элемента, с которым работает цикл, должен соответствовать типу массива. Можно использовать var. С кортежами не работает.

Foreach (var element in myArr)

{

Console.WriteLine(element);

}

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

Кортеж — упорядоченный набор фиксированной длины.

Кортежи предоставляют удобный способ для работы с набором значений

1. **Что такое локальная функция? Какова область ее видимости?**

Локальные функции представляют функции, определенные внутри других методов.

1. **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.

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

uncheked

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

Fixed - Ключевое слово позволяет «закрепить» локальную систему в стеке, чтобы предотвратить их сбор или перемещение во время сборки мусора. Он используется для сценариев низкого уровня программирования.