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

**Net Framework** – платформа, которая создана ***Microsoft*** для разработки приложений.

* общеязыковая исполняющая среда (CLR) Common Language Runtime) Виртуальная Машина. Обеспечивает выполнение сборки (управление памятью, загрузка сборок, безопасность, обработка исключений, синхронизация)
* библиотека классов (FCL). (.NET Framework Class Library) спецификации объектно-ориентированной библиотеки классов, интерфейсов и системы типов (типов-значений)соответствующая CLS
  1. Что такое CLR, FCL/BCL, CLI, IL?

CLR **(Common Language Runtime)**: — это **исполняющая среда** в .NET, которая управляет выполнением программ, написанных на различных языках программирования, совместимых с .NET

**CLI (Common Language Infrastructure):**

CLI — это стандарт, описывающий:

* **Общие типы данных** и **общую среду выполнения** для программ на разных языках программирования.

IL (или **CIL — Common Intermediate Language**) — это промежуточный язык, в который компилируются программы перед выполнением. Когда код компилируется, он переводится в IL, который затем исполняется через JIT-компилятор.

FCL и BCL представляют собой библиотеку классов в .NET Framework.

* **FCL** Framework Class Library — это большая библиотека классов, предоставляющая доступ ко множеству функций .NET, таких как работа с файлами, сетью, графическим интерфейсом, базами данных и др.
* **BCL** Base Class Library — это подмножество FCL, которое включает **базовые классы**, такие как классы для работы с коллекциями, строками, числами, потоками ввода-вывода и др.
  1. Пояснить работу JIT-компилятора?

 **Компиляция в IL**: Исходный код программы сначала компилируется в IL (промежуточный язык), а не в машинный код.

 **JIT-компиляция**: Когда программа начинает выполняться, JIT компилирует IL в **машинный код** для конкретного процессора.

* Это происходит **по требованию** — когда метод или часть программы вызывается, JIT-компилятор преобразует его в машинный код.

 **Исполнение машинного кода**: Скомпилированный машинный код затем исполняется на процессоре.

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

CTS (Common Type Systems)- спецификация типов, которые должны поддерживаться всеми языками ориентированными на CLR

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

**System.Object** — это **базовый класс для всех типов в .NET**. Этот класс предоставляет базовое поведение для всех объектов. Тип **System.Object** определяет следующие методы:

**ToString()** — возвращает строковое представление объекта.

**Equals(object obj)** — определяет, равен ли текущий объект другому объекту.

**GetHashCode()** — возвращает хеш-код объекта, который может использоваться в алгоритмах хеширования.

**GetType()** — возвращает тип объекта во время выполнения.

* 1. Что находится в mscorlib dll?

В mscorlib.dll находятся основные классы .NET, включая типы данных, исключения и базовые классы для работы с объектами.

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

*Сборка* (assembly) — это абстрактное понятие, для логической группировки одного или нескольких управляемых модулей или файлов ресурсов.

Сборка .NET состоит из манифеста сборки, метаданных и кода. Набор [ресурсов](https://learn.microsoft.com/ru-ru/dotnet/core/extensions/resources)

* 1. Какие виды сборок существуют?

с нестрогими именами (weakly named assemblies)

со строгими именами (strongly named assemblies).

* 1. Что такое assembly manifest?

**Манифест** - ключевой компонент сборки.

Он хранит в себе:

* Имя сборки
* Номер версии
* Язык и региональные стандарты
* Список всех файлов сборки
* и т.д
  1. Что такое GAC?

**GAC (Global Assembly Cache)** — В глобальном кэше сборок сохраняются сборки, специально предназначенные для совместного использования на компьютере несколькими приложениями.

* 1. Чем managed code отличается от unmanaged code

 **Managed code** управляется CLR, с автоматическим управлением памятью и безопасностью.

 **Unmanaged code** исполняется непосредственно операционной системой, требует явного управления памятью.

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

**Метод Main** — это точка входа в программу .NET. Определяет начало выполнения приложения.

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

 **using** для подключения пространств имен.

 **using** для объявления области видимости для объектов (управление ресурсами).

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

Сборки могут содержать несколько пространств имен. Пространства имен организуют код внутри сборки, помогая избежать конфликтов имен.

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

**Примитивные типы данных** (или **значимые типы данных**) — это базовые типы данных, которые предоставляются языком программирования для хранения простых значений. (значимые типы): int, float, double, char, bool, byte, short, long, uint, ulong, ushort.

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

**Ссылочные типы** хранят ссылку на объект в памяти. Относятся к ним: string, class, object, delegate, dynamic, array.

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

**Типы-значения** включают: int, float, double, char, bool, byte, short, long, uint, ulong, ushort, sbyte.

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

**Значимые типы** хранят данные непосредственно. **Ссылочные типы** хранят ссылку на объект в памяти.

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

**Упаковка** (boxing) — преобразование значимого типа в ссылочный тип (object). **Распаковка** (unboxing) — преобразование ссылочного типа обратно в значимый.

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

**int** и **System.Int32** (а также double и System.Double) — это два способа представления одного и того же типа. Первый — это **псевдоним** (alias) для второго.

* 1. Для чего используется тип dynamic?

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 var нельзя менять тип данных

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

Переменная, объявленная с помощью **var**, где тип выводится на основе присвоенного значения.

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

Чтобы можно было присваивать переменной null-значение

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

Строковый литерал объявляется в двойных кавычках: "example". Можно выполнять операции: конкатенация, извлечение подстроки, поиск, замена, форматирование. Сравнение

-Поиск

-Разделение

-Обрезка

-Вставка

-Удаление

-Замена

-Смена регистра

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

Строки можно задавать с помощью строковых литералов, конкатенации, использования методов класса String, или через интерполяцию строк.

* 1. Какие методы есть у типа String?

сцепление, копирование, выделение подстроки, разделение строки на слова, вставки подстроки в заданную позицию, удаление заданной подстроки

* 1. В чем отличие пустой и null строки?

**Пустая строка** ("") имеет длину 0, но существует. **Null строка** (null) не инициализирована и не указывает на объект.

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

Используя метод String.Compare или операторы == и != для сравнения строк.

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

**String** — неизменяемый тип, каждое изменение создает новый объект. **StringBuilder** — изменяемый тип, позволяет изменять содержимое строки без создания новых объектов.

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

Методы **Convert.ToInt32()**, **Convert.ToDouble()** и т.д. позволяют преобразовывать значения между типами (например, строку в число)

* 1. Как выполнить консольный ввод/вывод?

Для **вывода** используйте Console.WriteLine(). Для **ввода** используйте Console.ReadLine().

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

 **Одномерный массив**:

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

 **Двумерный массив**:

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

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

**Ступенчатый массив** — это массив массивов с разными длинами. Он задается как int[][] с разными размерами вложенных массивов.

* 1. **Какие типы можно использовать в foreach**? Приведите пример.
  2. Что такое кортеж? Для чего и как он используется?

**Кортеж** — это структура данных, позволяющая объединить несколько значений разных типов в одном объекте.

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

**Локальная функция** — это функция, объявленная внутри другого метода. Она доступна только в пределах метода, где была объявлена.

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

 **checked** — включает проверку переполнения для арифметических операций.

 **unchecked** — отключает проверку переполнения

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

По умолчанию, арифметические операции в C# не проверяются на переполнение (непроверенный контекст). Можно явно указать checked или unchecked блок для изменения поведения.

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

**fixed** используется для предотвращения перемещения объекта в памяти сборщиком мусора, что необходимо при работе с указателями. **Особенности**: может использоваться только в небезопасном коде (unsafe context).