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

Программная платформа обеспечивает совместное использование разных языков программирования + безопастность; состоит из CLR (общеязыковая среда), MSIL (Microsoft CIL & CLI), FCL (Microsoft CLI BCL оо библиотека классов)

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

CLR [common language runtime] (общеязыковая среда), FCL [framework class library] (оо библиотека классов, интерфейсов, систем типов), BCL [base class library] (библиотеки и профили), CLI [common language infrastructure] (спецификация общеяз структуры, определяет архитектуру исполнительной систему и набор предоставляемых сервисов), CIL [common intermediate language] (промежуточный язык), IL (объектно-ориентированный машинный язык не зависящий от процессора)

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

Оперативная компиляция:

1)CLR ищет типы данных и загружает во внутренние структуры

2)Для каждого метода CLR заносит адрес внутренней CLR функции

3)JIT Compiler ищет в метаданных сборки IL-код вызываемого метода, проверяет и компилирует IL-код в машинные команды

4)Они хранятся в динамически выделенном блоке памяти.

5)JIT Compiler заменяет адрес вызываемого метода адресом блока памяти, содержащего готовые машинные команды

6)JIT Compiler передает управление коду в этом блоке памяти.

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

Идет из CLS (общеязыковая спецификация) CTS (Common Type Systems) – спецификация типов, которые должны поддерживаться всеми языками ориентированными на CLR.

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

Все типы и классы принадлежат System.Object -> могут реализовать те методы, которые определены в классе System.Object.

**6. Что находится в mscorlib dll?**

mscorlib.ni.dll – это файл DLL в Windows. DLL – это аббревиатура от Dynamic Link Library (динамически подключаемая библиотека). DLL файлы необходимы программам или расширениям веб-браузера, потому что они содержат программный код, данные и ресурсы.

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

Сборка – это логическая единица (portable executable [исполняемый файла (*EXE*) + файла библиотеки динамической компоновки (*DLL)*]), содержащая скомпилированный код, используемый внутри CLR /логическая группировка управляемых модулей или файлов ресурсов.

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

8. Какие виды сборок существуют?

**9. Что такое assembly manifest?**

Манифест – набор таблиц метаданных.

Манифест сборки (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, а также набор прав доступа, которые необходимы сборке для корректной работы.

**10. Что такое GAC?**

Global Assembly Cache — подсистема, хранящая сборки CLI в централизованном репозитории.

**11. Чем managed code отличается от unmanaged code**

*managed code* — код программы, исполняемой под «управлением» виртуальной машины .NET — Common Language Runtime или Mono. При этом машинный код называется *неуправля́емым кодом* (*unmanaged code*).

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

Каждая программа на C# должна иметь один метод Main (c большой буквы). Данный метод часто называют точкой входа в программу.

static void Main()

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

usingSomeUtil=MyCompany.Utilities.SomeUtility;//псевдоним

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

Различные типы, принадлежащие одному пространству имен, могут быть реализованы в разных сборках.

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

Примитивные типы - тип данных, предоставляемый языком программирования как базовая встроенная единица языка:

* Целочисленные со знаком (sbyte, short, int, long)
* Целочисленные без знака (byte, ushort, uint, ulong)
* IEEE-представление с плавающей запятой (float, double)
* Десятичный с повышенной точностью (decimal)
* String
* Object

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

Ссылочные типы хранятся в куче или хипе, которую можно представить как неупорядоченный набор разнородных объектов

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

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

* Целочисленные типы (byte, sbyte, short, ushort, int, uint, long, ulong)
* Типы с плавающей запятой (float, double)
* Тип decimal
* Тип bool
* Тип char
* Перечисления enum
* Структуры (struct)

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

Значения типов-значений хранятся в стеке (растет снизу вверх: каждый новый добавляемый элемент помещается поверх предыдущего); ссылочные – в куче (неупорядоченный набор разнородных объектов): в стеке помещается ссылка на адрес в куче (хипе). Когда объект ссылочного типа перестает использоваться, в дело вступает автоматический сборщик мусора: он видит, что на объект в хипе нету больше ссылок, условно удаляет этот объект и очищает память - фактически помечает, что данный сегмент памяти может быть использован для хранения других данных.

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

Упаковка – преобразование типа значения в ссылочный тип. Используется для хранения типов значений в куче со сбором мусора.

Распаковка – преобразование из ссылочного типа в тип значения.

Упаковка является неявной; распаковка является явной.

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

Никакого, просто другое более короткое название.

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

Может получить какое угодно начальное значение, и на протяжении времени его существования это значение может быть заменено новым

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

После присваивания в var значение изменить нельзя в dynamic можно

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

Var – переменная в которой компилятор сам решает к какому типу он относится. Должно сразу инициализироваться.

24. Для чего используют Nullable тип?

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

Console.WriteLine("Компания \"Рога и копыта\"");

string s3 = s1 + " " + s2; // конкатенация

string s10 = string.Join(" ", values); //объединение

int result = string.Compare(s1, s2); // сравнение (0-идентичны, <0-первая перед второй, >0-вторая перед первой)

int indexOfChar = s1.IndexOf(ch); // индекс первого вхождения

LastIndexOf, StartsWith и EndsWith

string[] words = text.Split(new char[] { ' ' }); // разделение строки на массив подстрок

text = text.Trim(new char[] { 'd', 'h' }); // обрезка строки

text = text.Substring(2); / text = text.Substring(0, text.Length - 2); //обрезать часть строки

text = text.Insert(8, substring); //вставка

text = text.Remove(0,2); //удалить часть строки

text = text.Replace("хороший", "плохой");//замена

Console.WriteLine(hello.ToLower()); / ToUpper //замена регистра

**26. 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' });

string s4 = new String(new char[] { 'w', 'o', 'r', 'l', 'd' }, 1, 3); // orl

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

Compare, Contains, Concat, CopyTo, EndsWith, IndexOf, LastIndexOf, Insert, Join, Replace, Split, Substring, ToLower, ToUpper, Trim

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

NULL – это тип данных, который используется для обозначения пустой переменной. А строка с пустым значением инициализируется как тип string.

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

int MyInt = string1.CompareTo(string2); //(0-идентичны, <0-первая перед второй, >0-вторая перед первой)

CompareTo с учетом регистра

Compare

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

String неизменяемая строка, в отличие от StringBuilder (динамическая строка)

31. Поясните явные преобразования переменных с помощью команд Convert.

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

Console.WriteLine(), Console.Write(), Console.ReadLine(), Console.ReadKey(), Console.ReadLine()

33. Приведите примеры определения и инициализации одномерных и двумерных массивов.

Int[] arr=new int[10];

Int [,] arr=new int[2,3]

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

Int [][] arr=new int [2][] //на 2 строки или int[][] arr={new int[4],new int[3],}

Arr[0]=new int [4] // первая строка из 4 элементов

…

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

Перебор элементов в контейнерах. Работает только на чтение (наполнять нельзя)

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

foreach(inti innumbers)

{ Console.WriteLine(i); }

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

Кортеж – тип данных, который комбинирует объекты разных типов (1-8). Не изменяется, только для чтения

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

Локальная ф-я – ф-я определенная внутри других методов. Область видимости л.ф внутри класса – класс; внутри др. функции – внешняя фуныкция.

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

Checked проверяется на переполнение -> используется метод try-catch-finish, где при встрече переполнения, то генерируется исключение *OverflowException* и выполняется блок catch

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

По умолчанию unchecked. byte c = checked((byte)(a + b));

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

Классы (ссылочные типы) помещают значения в куче -> сборщик мусора может уничтожить значения -> для фиксации указателей на время работы используется оператор fixed

Объявленный указатель можно использовать только в соответствующей fixed инструкции. Объявленный указатель доступен только для чтения и не может быть изменен.

unsafe

{

var message = "Hello!";

fixed (char\* p = message)

{

Console.WriteLine(\*p); // output: H

}

}