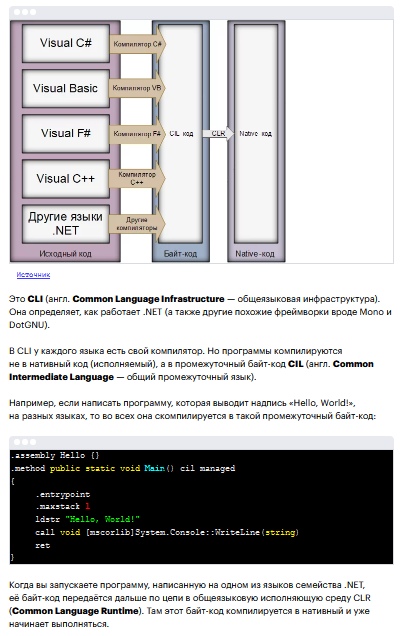
1. Что такое .Net Framework и из чего он состоит?

.NET — это фреймворк от Microsoft, который позволяет использовать одни и те же пространства имён, библиотеки и API для разных языков. Чаще всего это четыре языка из семейства .NET



2. Что такое CLR, FCL/BCL, CLI, IL?

CLR (Common Language Runtime) – Среда Времени Выполнения или Виртуальная Машина. Обеспечивает выполнение сборки (управление памятью, загрузка сборок, безопасность, обработка исключений, синхронизация)

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

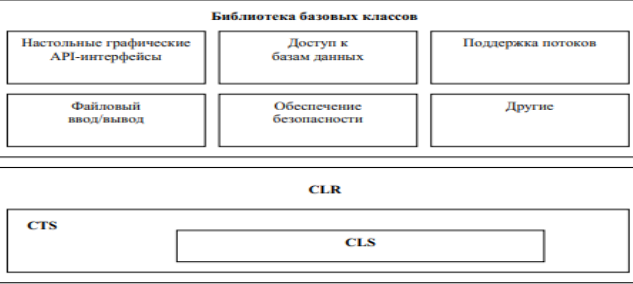
CLI (Common Language Infrastructure) – спецификация общеязыковой инфраструктуры.

Определяет архитектуру исполнительной системы и набор представляемых сервисов.

Стандарты: ECMA-335 и ISO/IEC 23271.

IL объектно-ориентированный машинный язык не зависящий от процессора.

- intermediate language - код,содержащий набор инструкций , не зависящих от платформы .иными словами , после компиляции исходного кода он преобразует не в код для платформы , а в промежуточный код на языке il.



МЕ Т-при ложен ИЯ 
Базовые типы: 
Windows 
АррисаНоп, 
Console 
Application Class 
Library 
Традиционные 
Wind0WS- 
гтриложения 
VisuaI Studio ЛЕТ 
Языки программи;ювания Microsoft (VB, С!С++, 
JSchpt) и независимых поставщиков 
Ct:vnmon [алдиаде Speti6cabon 
Типы МЕТ-приложений (ConsoIe, Whdows Forms, 
ComponerMs, ASP „НЕТ, Web и пр.) 
.МЕТ Framework 
Библиотеки клхсов 
CLR (Соттоп Language Runtime) 
Wind0WS 
Сервиаы опе;вционеюй системы (Wjn32 АА) 

3. Пояснить работу JIT-компилятора?

JIT-компиляция (англ. Just-in-time compilation, компиляция «на лету»), динамическая компиляция (англ. dynamic translation) — технология увеличения производительности программных систем, использующих байт-код, путём компиляции байт-кода в исполняемый непосредственно во время работы программы.

Когда среда CLR выполняет IL код, то происходит следующее: при вызове блока IL-кода JIT-компилятор проверяет и преобразует его в машинные команды которые он сохраняет в динамическом блоке памяти. Затем, он возвращается к структуре данных типа и заменяет адрес вызываемого метода адресом этого блока памяти и при следующем запросе повторная компиляция не будет выполнена, а будет использован уже скомпилированный код.

CLR (Соттоп Language Runtime — 
общеязыковая исполняющая среда) 
оперативная 
компиляция 
(just In Time 
- лт). 
машинный код 

машинный 
јп-код 
компилятора 
Оптимизация 
/optimize и /debug 
NGen.exe код 

ЈIТ-компилятор (Just-In-Time) 
1) 
2) 
3) 
4) 
5) 
6) 
CLR ищет типы данных и загружает во внутренние 
структуры 
Для каждого метода CLR заносит адрес внутренней 
CLR функции jITCompiler 
jITCompiler ищет в метаданных соответствующей 
сборки вызываемого метода, проверяет и 
компилирует в машинные команды 
они хранятся в динамически выделенном блоке 
памяти. 
jITCompiler заменяет адрес вызываемого метода 
адресом блока памяти, содержащего готовые 
машинные команды 
JITCompiler передает управление коду в этом блоке 
памяти. 

4. Что такое CTS (Common Type System)?

Common Type System (CTS) is a standard that specifies how type definitions and specific values of types are represented in computer memory

Система общих типов (CTS) - это стандарт, который определяет, как определения типов и конкретные значения типов представлены в памяти компьютера.

5. Какие аспекты поведения определяет тип System.Object? Он является исходным базовым классом для всех классов платформы .NET Framework и корнем иерархии типов.

6. Что находится в mscorlib dll? mscorlib определенно является одной из библиотек базового класса .net, и каждая программа на С# зависит от нее.

**В этой сборке содержится большое количество базовых типов, охватывающих широкий спектр типичных задач программирования, а также базовых типов данных, применяемых во всех языках .NET**

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

Сборка (assembly) — 1) это абстрактное понятие, для логической группировки одного или нескольких управляемых

модулей или файлов ресурсов.

2) дискретная единица многократно используемого кода внутри CLR

Exe, dll

8. Какие виды сборок существуют?

►с нестрогими именами (weakly named

assemblies)

► со строгими именами (strongly named

assemblies).

§ подписаны при помощи пары ключей,

уникально идентифицирующей издателя

сборки (безопасность, управление ее

версиями, развертывание в любом месте

пользовательского жесткого диска или в

Интернете)

§ атрибуты: имя файла (без расширения), номер

версии, идентификатор регионального стандарта и

открытый ключ.

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

Все эти компоненты могут находиться в одном файле, и тогда сборка представляет один единственный файл в формате exe или dll.

9. 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.

10.Что такое GAC?

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

11.Чем managed code отличается от unmanaged code. В общем случае при написании С#-программы создается код, называемый управляемым (managed code). Управляемый код выполняется под управлением CLR-системы. Неуправляемый код - это обычный машинный код. Практически все программы написанные не на базе .NET используют неуправляемый код.

12.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.

13.Варианты использования директивы using( using Directive ) в C#. (мб чтоб исп. Сборки и пространства имен?)

разрешает использование типов в пространстве имен, поэтому уточнение использования типа в этом пространстве имен не требуется

using System.Text;

позволяет создавать псевдонимы пространства имен или типа.Это называется директива using alias.

14.Как связаны между собой сборки и пространства имен? - Может нужн и там, и там использоваьь using перед использванием.

15.Типы данных, которые поддерживаются компилятором напрямую, называются примитивными (primitive types) или

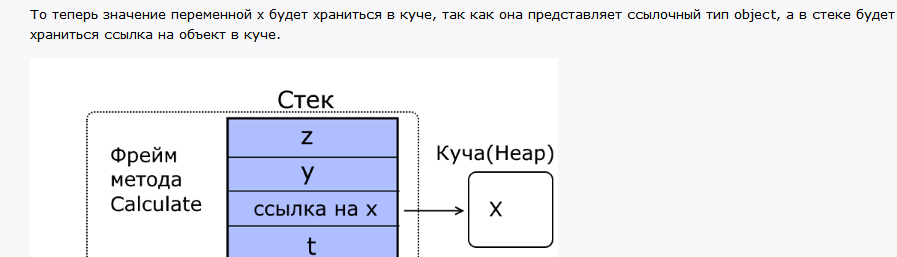
встроенными у них существуют прямые аналоги в библиотеке классов .NET Framework Class Library

BCT 
short 
int 
Ion 
ushort 
uint 
char 
bml 
float 
d(Nble 
decimal 
eHHble 
Pa3Mep 
B 6max 
8 
16 
32 
64 
8 
16 
32 
64 
16 
8 
32 
64 
128 
n VIM1,1TMBHble 
Thn (ECL) 
S 
stem.Int16 
S 
em.Int32 
em.Int64 
S stem B 
S 
stem.Uint16 
S 
stem.Uint32 
S 
em.Uint64 
em.Char 
stem.Boolean 
S 
stem.Si le 
S 
stem.Double 
S 
em.Decimal 
em.Stri 
TV1nbl C# 
m 29 
I.oxlo-z 

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

CCb•nOMHb1e Thnbl 
ing obj Something; 
CLR Tpe6yeT, qT06bl BCe 06beKTbl c03Aaænncb 
HeT del&e, TO ecTb RB»-0ro cnocc6a naMAT", 
3.aHqToi cßbeKTOM. Y&IPKOÄ Myco* 3aHMMaeTcß cpeaa CLR 
Something obj Something; 
objSomething = new Something() ; 
MMH (nceBAOHHM) 
stri ng 
Tun CTS 
System.String 

» Ссылочный тип 0bjegt 
» В CLR каждый объект прямо или косвенно 
является производным от System.Object 
/ / Тип, неявно производный от 0bject 
c1ass Student 
/ Тип, явно производный от 0bject 
c1ass Person : System.Object 



Где выделяется память

под ссылочные типы?

17.Какие типы относятся к типам-значениям?

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

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

19.Что такое упаковка и распаковка значимых типов?

» Упаковкой (boxing) называется процесс преобразования 
типа значения в тип System.Object или в тип 
интерфейса, который реализуется данным типом- 
значением 
Int32 х „ 5; 
()bject о = х; 
1. з угравляеной ПТ*ЯТь 
2. гюля копияются 
З. возвращается адрес (Кьекта 
/ / упаковка х; О ссылается на упакованный объект 
» распаковка (unboxing) 
получение указателя на исходный значимый тип (поля 
данных), содержащийся в объекте 
Сбекты значимого типа существуют в двух фрмах: 
(unboxed) и упакоин»юй (tmed). Ссытчнье т»ты бывают только в 
упиоваю-юй форме. 

int х 5; 
0bject о = х; 
byte т 
Н Упаковка х 
Генерируется InvalidCastException 
1. Если (Креаенная, содержащая ссылку на 
упакованный значимый тип, равна 
пи“, генерируется исключачие 
NuIIReferenc±xception. 
2. Если ссылка указывает на объект, не 
являющийся упакованным значением 
тр±уемого значиьюго типа, генерируется 
исключение InvalidCastException 
int х „ 
()bject о х; упаковка х 
byte т • / / Распаковка, 
а Затем приведение типа 

class lection 
{ struct A 
y rEKOBKa 
04 na.0T 
n oxeHHß 
{ public int Val; } 
public static void Main() 
A myA new A(); 
myA.VaI = 5 
object refType = myA; // ynaK0BKa 
A ValType2 = (A) refType; // pacnaKOBKa 
npneæ-we æynaK0BaH*-oro 3K3ænngpa »eqnoro TMna K 
OAHOMY "3 " RTeXEäcoB 3Toro Tma TpéyeT, 
Sun ynaK0ea-. 

Назначение:

►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,

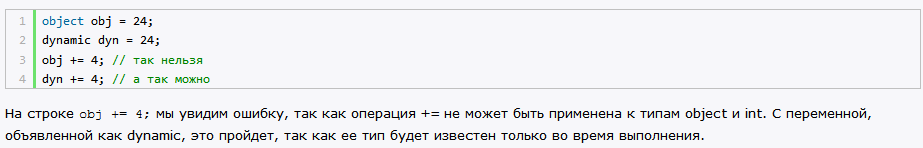
методы Object.

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

int это ключевое слово для c#, ровно как Integer для VisualBasic, но оба они представляют собой System. Int32 . int - примитив, доступный в компиляторе C#, в то время как Int32 является типом FCL и следовательно доступен для всех языков, соответствующих CLS.

1. Для чего используется тип dynamic?

объявленные как dynamic, могут в течение работы программы менять свой тип



**dynamic это тот же object, просто игнорирует все проверки на тип во время компиляции.**

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

dynamic – Тип которому может быть присвоено любое значение и в дальнейшем может быть изменено, var не изменить

23.Что такое неявно типизированная переменная?

Неявно типизированная переменная объявляется с помощью ключевого слова var и должна быть непременно инициализирована

24.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

int 
int? 
х = пит; 
= пи11; 
х 
упрощенная форма использования 
структуры 
которая позволяет пил значения 
применяется только для Типов значении 

25.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

Конкатенация

26.Какие есть способы для задания и инициализации строк?

* Из массива символов

char[] chararray = {'e', 'x', 'a', 'm', 'p', 'l', 'e'};  
 string str = new string(chararray);

* Инициализация отложена (потом присвоим значение)
* Инициализация строковым литералом

string s = "This is my stroke";

* С помощью String builder

27.Какие методы есть у типа String?

Compare: сравнивает две строки с учетом текущей культуры (локали) пользователя

· 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

* Format Форматирование в соответствии с заданными
* спецификаторами формата
* IndexOf,
* LastIndexOf, …
* Определение индексов первого и последнего вхождения
* заданной подстроки или любого символа из заданного
* набора
* Insert Вставка подстроки в заданную позицию
* Join Слияние массива строк в единую строку. Между
* элементами массива вставляются разделители (см.
* далее)
* Length Длина строки (количество символов)
* Remove Удаление подстроки из заданной позиции
* Replace Замена всех вхождений заданной подстроки или символа
* новой подстрокой или символом
* Split Разделение строки на элементы, используя заданные
* разделители. Результаты помещаются в массив строк
* Substring Выделение подстроки, начиная с заданной позиции

28.В чем отличие пустой и null строки?

► Пустая строка — экземпляр объекта System.String, содержащий 0 символов:string s = "";

Для пустых строк можно вызывать методы.

► Строки со значениями null не ссылаются на экземпляр объекта System.String, попытка вызвать метод для строки null вызовет исключение NullReferenceException. строки null можно использовать в операциях объединения и сравнения с другими строками.

29.Как можно выполнить сравнение строк?

* Compare Сравнение двух строк в алфавитном порядке. Разные реализации метода позволяют сравнивать строки и подстроки с учетом и без учета регистра и особенностей национального представления дат и т. д.
* CompareOrdinal Сравнение двух строк по кодам символов. Разные реализации метода позволяют сравнивать строки и подстроки
* CompareTo Сравнение текущего экземпляра строки с другой строкой

30.В чем отличие типов String и StringBuilder?

Любые модификации строки происходят внутри блока памяти

§ Length - длина строки

§ Capacity - максимальная длина строки,

Основное отличие - StringBuilder можно изменять после создания.

Строка иммьютабельна (её невозможно изменить после того, как она созадана). Если вы что-то прибавляете к строке, то в действительности вы не получите старую строку + добавку, вы получите новую страку, составленную из старой и добавки. Отсюда все проблемы с производительностью в приложениях, которые последовательно формируют большие строки.

StringBuilder же будет последовательно выделять память и "по-настоящему" дозаписываться данными. Так что при построении больших строк - только StringBuilder.

31.Поясните явные преобразования переменных с помощью команд Convert.

Существует ещё один способ преобразования данных с помощью класса Convert, в котором есть много статических методов (с префиксом To). Данные статические методы выполняют преобразования.

Convert.ToByte(Var1), Convert.ToChar(Var1), Convert.ToDouble(Var1).

32.Как выполнить консольный ввод/вывод?

статический класс System.Console.

Методы класса Console

• Clear: очистка консоли

• WriteLine: вывод строки текста с переводом на новую строку

• Write: вывод строки текста

• Read: считывание введенного символа в виде числового кода данного символа. С помощью преобразования к типу char мы можем получить введенный символ

• ReadKey: считывание нажатой клавиши клавиатуры (ConsoleKeyInfo key= Console.ReadKey();) • ReadLine: считывание строки текста со входного потока

33.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 } };

Одномерные массивы 
(single-dimensional) 
тип[] имя; 
тип[] имя = new тип [ размерность ]; 
тип[] имя = { список инициализаторов }; 
тип[] имя = new тип [1 { список 
инициализаторов }; 
тип[] имя = new тип [ размерность ] { 
список инициализаторов }; 
int[] 
int[) 
int[] 
int[) 
int[] 
а; 
Ь; 
{ 61, 2, 5, -9 у; 
d = new int[] { б, 2, 5, -9 у; 
е = пен int[4] { 61, 2, 5, -9 
int[) е = пен int[4] { 61, 2, 5, -9 у; 
foceach ( int х in е ) Console.Wr•iteLine 

Прямоугольные (многомерные) 
массивы 
тип[,] имя; 
тип[,] имя = new тип [ разм_1, разм_2 ]; 
тип[,] имя = { список инициализаторов }; 
тип[,] имя = new тип [,] { список инициализаторов }; 
тип[,] имя = new тип [ разм_1, разм_2 ] { список 
инициализаторов }; 
iat[,] е; //•лементов нет 
с • 2, З), {4, 5, // пен подразу—вается 
f 2, {4, 5, мчисл"тся 
2, 3), {4, 5, олисание 
b[i, ј] 
int[, 
foreach (vap х in а) 
Console . ••\t" 

34.Что такое ступенчатый массив? Как его задать?

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

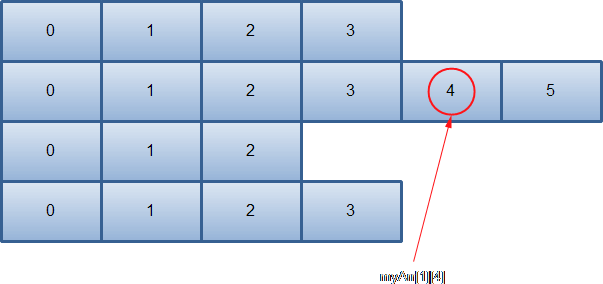
myArr[0] = new int[4];

myArr[1] = new int[6];

myArr[2] = new int[3];

myArr[3] = new int[4];

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



CTyneHqaTble,3y6qaTble 
jagged MaCCVlBbl 
foreach 
a { new int[3], new int[5], new int[4] 
(int[] x in a) 
foreach (int b in x) 
Console + b); 
Console.WriteLine(); 
1,1Mq; 
a • new // 
all] = new int[3); // nay.AT" 
.12] • new intL4J; // WAeney•e 
nOA 
cypoKy (5 
noA 1-0 cypo«y (3 
noA 2-n (4 
snemeHtoe) 
•nemeHTa) 
{ new int(S), new int[3], neu int[4] 
Ann npoaonxeHHR HaxMHTe 
n06yE 
Knagvuy . . 

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

» цикл foreach 
Цикл foreach приназначен для 
перебора элементов в контейнерах, 
в том числе в массивах. 
' tT1ng[ 
collettion 
foreach (тип_иден. название_иден. in контейнер) 
{ операторы 
цикл работает только на чтение, но не на 
запись элементов (наполнять нельзя) 
Д.б. интерфейса ЕпипЕгаЫе 
int[] ПипЈе'% = пех int[] 1, 2, З 4 
foreach (int in numbers) 
Сопзо1е .WiteLine(i) ; 
Stcing[] 
-1-й элемент“, 
“2-В элемент“, 
«3-й элемент“ 
/ /Последовательно выводим в консоль массива 
foceach (Steing e1ement со11есИоп) 
[Цикл for, 
for• (int i 
Cons о1е . ; 
њтолнякций аналогич. действия 
• В; i C011ection.Length; +4i) 
Console i]) ; 

numbers • new 
numberstø] new int(] { l' 
numbers[l] new int(l { I, 
numbers[2] • new int(] { 1, 
foreach (inta in numbers) 
2}; 
foreach (int number in rcv) 
Console.EriteLine(); 

1 StringBuilder объекты

2 String и string переменные

3 Массивы

4 Классы с реализованным интерфейсом IEnumerable

5 Коллекции

foreach (var item in sb1)

{

Console.WriteLine(item);

}

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

Кортежи (tuple) комбинируют объекты различных типов (от одного до восьми). Типы и выражения.

Кортеж — упорядоченный набор фиксированной длины.

Кортежи предоставляют удобный способ для работы с набором значений

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

Локальные функции представляют функции, определенные внутри других функций.

может быть вызвана только внутри этого функции(метода)

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

В проверенном контексте арифметическое переполнение создает исключение. В непроверяемом контексте арифметическое переполнение игнорируется и результат усекается путем удаления старших разрядов, которые не помещаются в целевой тип данных

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

Checked

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

Язык C# предоставляет специальное ключевое слово fixed, чтобы указать сборщику мусора не перемещать объект. Это означает, что позиция переменной в памяти в памяти фиксируется, чтобы на нее мог ссылаться указатель. На C# это называется прикреплением (pinning).

Кроме адреса переменной можно также инициализировать указатель, используя массив, строку или буфер фиксированного размера

fixed(int\* p = nums)

{

При инициализации указателей на строку следует учитывать, что указатель должен иметь тип char\*.