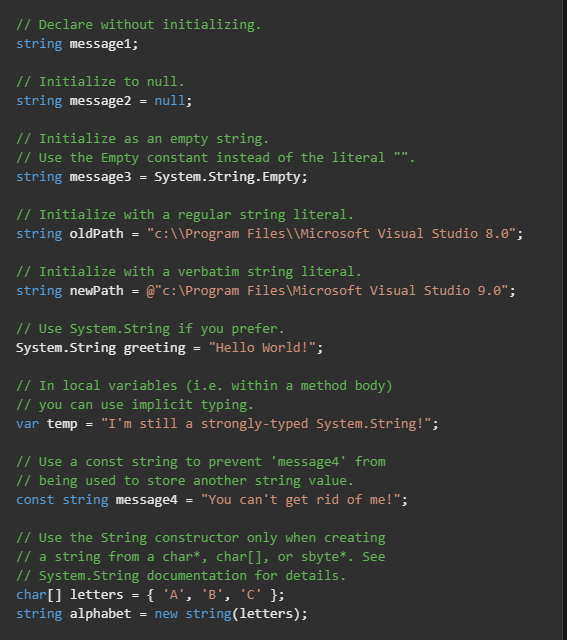
1. **Что такое .Net Framework и из чего он состоит?-** CLR (Common Language Runtime) – Среда Времени Выполнения или Виртуальная Машина. Обеспечивает выполнение сборки (управление памятью, загрузка сборок, безопасность, обработка исключений, синхронизация) ► FCL (.NET Framework Class Library) – соответствующая CLS спецификации объектно-ориентированная библиотека классов, интерфейсов и системы типов (типов-значений)
2. Что такое CLR, FCL/BCL, CLI, IL? – CLR-общеязыковая среда выполнения(предоставляет среду и выполнение для разных языков) FCL-библиотека классов Framework BCL- библиотека базовых классов CLI-Общая языковая инфраструктура(CLI - это спецификация, а CLR - реализация этой спецификации.) IL-(Instruction List) представляет собой текстовый язык программирования низкого уровня, который очень похож на Assembler
3. Компиляция во время выполнения
4. Система общих типов CTS определяет способ объявления, использования и управления типами в среде CLR, а также является важной составной частью поддержки межъязыковой интеграции в среде выполнения.
5. System.Object — класс в C#, который является базовым для всех типов данных (как структурных, так и ссылочных). Это класс определяет общее полиморфическое поведение для всех типов в мире .NET. System.Object Не обязательно указывать в программе, как базовый — это подразумевается само собой. Однако ничто не мешает это сделать
6. Mscorlib.dll это файл библиотеки динамических ссылок, который является частью Microsoft® Silverlight
7. 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 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.
8. уществует два вида сборок: однофайловые сборки; многофайловые сборки. Сборка, которая состоит из одного единого модуля (\*.dll или \*.exe) называется однофайловой. В однофайловых сборках все необходимые CIL -инструкции, метаданные и манифесты размещаются в одном, четко определенном пакете. Сборка, которая состоит из многих файлов двоичного кода .NET, называется многофайловой. Каждый из этих файлов называется модулем.
9. **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.
10. **подсистема, хранящая сборки CLI (CLI assembly) в централизованном репозитории**. Система разработана для борьбы с проблемами разделяемых библиотек, например таких, как DLL hell.
11. **Managed code** -компилируется не в машинный код,а в промежуточный и работает в рамках управляемой среды. Unmanaged code -код компилируется в машинный код и обрабатывается ОС
12. Использование функции main () обязательно в таких языках, как Java, потому что это упрощает понимание того, в каком порядке код запускается в программе
13. Директива using позволяет использовать типы, определенные в пространстве имен, без указания полного пространства имен этого типа. В базовой форме директива using импортирует все типы из одного пространства имен
14. Между сборками и пространствами имен нет прямого соответствия: в сборке может хранится несколько пространств имен, а разные классы одного пространства имен могут быть определены в разных сборках.
15. **примитивные типы данных** представляют собой набор базовых [типов данных](https://translated.turbopages.org/proxy_u/en-ru.ru.2991af25-631b352f-3cf6f00b-74722d776562/https/en.wikipedia.org/wiki/Data_type" \o "Тип данных), из которых создаются все остальные типы данных. **Строка, Число, Дата, Булево и другие**
16. **Ссылочные** **типы** данных ещё называют **ссылками**. **К** **ним** **относятся** все классы, интерфейсы, массивы, а также **тип** данных String.
17. **К** **типам**-**значениям** принадлежат переменные базовых **типов** (int, double, char и другие), структурные переменные (struct) и перечисления (enum)
18. **Это просто**
19. По лабе посмотрит
20. Нет разницы
21. **тип** переменной может измениться или что он неизвестен до выполнения
22. Данный оператор, как и **var**, позволяет вывести тип переменной исходя из присвоенного ей значения. Но в **отличие** от **var**, **dynamic** позволяет изменять тип переменной
23. **Неявно** **типизированные** **переменные** – это **переменные**, в месте создания которых вместо типа **переменной** пишется контекстное слово var и компилятор сам подставит необходимый тип данной **переменной**
24. Nullable value type (**тип значения, допускающий null**) – это тип, который позволяет представить не только все значения своего нижележащего типа, но и значение null.
25. Строковый литерал это кавычки. **Concat-объединение, Compare-сравнение, IndexOf** – поиск в строке, **Split** -разделение, **Insert-вставка.**



27)

28) **Пустая** **строка** - это **строка**, в которой нет символов. **NULL** - это вообще ничего. Если мы запросили перевод токена и получили **пустую** **строку**, значит токен непереводим. Если мы запросили перевод и результат - **NULL**, значит этим токеном никто не озаботился, и впору кидать эксепшен.

29) **Compare-сравнение**

33) int[] array = new int[5];

Массивы могут иметь несколько измерений. Например, следующее объявление создает двухмерный массив из четырех строк и двух столбцов.

int[,] array = new int[4, 2];

Следующее объявление создает массив из трех измерений: 4, 2 и 3.

int[,,] array1 = new int[4, 2, 3];

34)int[][] myArray = new int[3][]

40)fixed-закрепить значение в стеке