1. Вход исходного кода .NET совместимого языка -> проверка синтаксиса компилятором -> получение сборки(метаданные, манифест, CIL, ресурсы) -> VES получает сборку -> JIT кеширует(единоразово компилирует используемые методы) -> в результате формируется исполняемый модуль с машинными командами -> исполнение машинных команд процессором.
2. CLR — общеязыковая среда исполнения, которая позволяет преобразовывать код .NET совместимых языков в байт код и после в машинный код.
3. FCL — набор всех классов .NET платформы.
4. BCL — подмножество FCL, которое является минимальным требованием реализованных классов для того, чтоб язык программирования являлся .NET совместимым.
5. CTS — система описывающая как определяются и реализуются типы. Это позволяет реализовать взаимодействие между языками .NET-а.
6. CLS — подмножество CTS, на основе которой выбраны классы для BCL.
7. Единицей компиляции для VES является сборка.
8. JIT компиляция — это компиляция IL кода после обработки VES процессором. JIT компилирует методы чтоб в дальнейшем не проводит эту операцию повторно. Это называется кешированием. JIT компилятор на выходе формирует исполняемый модуль. При запуске исполняемого файла происходит компиляция из байт-кода в машинный код.
9. После проверки синтаксиса создается сборка, которая является по сути набором основных данных о типах, сборке, ресурсе и IL кода.
10. 4 части сборки:
    1. Метаданные — содержат информацию о классах(типах) присутствующих в исходном коде.
    2. Манифест — техническая информация о версии, локализации и другое о сборке.
    3. Ресурсы — подключенные файлы извне.
    4. CIL код — промежуточный объектно-ориентированный высокоуровневый байт-код содержащий в себе объекты.
11. IL-код — промежуточный объектно-ориентированный высокоуровневый байт-код содержащий в себе объекты. Это позволяет иметь общий “высокоуровневый ассемблер” независимо от используемого языка программирования. Также из IL-кода можно декомпилировать в язык совместимый с .NET.