

## 1. Система общих типов CTS. Что это такое?

CTS (стандартная система типов) — спецификация, которая определяет как какой-либо тип объявляется, используется и управляется в среде CLR. Как должны быть определены типы для того, чтобы они могли обслуживаться в CLR – среде. А так же является важной составной частью поддержки межязыковой интеграции в среде выполнения.

## 2. Цели CTS?

- Обеспечение возможности программ, написанным на различных языках программирования, легко обмениваться информацией
- Предотвращение определенного класса ошибок в программе до ее выполнения
- Осуществляет статическую проверку программы не только на уровне компилятора, но и на уровне системы выполнения

## 3. Какие функции выполняет CTS?

- CTS позволяет обеспечивать межязыковую интеграцию, безопасность типов и высокопроизводительное выполнение кода
- Обеспечивает ОО модель, которая поддерживает полную реализацию многих ЯП.
- Определяет правила, гарантирующие, что типы данных объектов, написанные на разных языках, смогут взаимодействовать друг с другом
- Предоставляет библиотеку, которая содержит типы примитивы
- Обеспечивает безопасную типизацию (корректное преобразование типов, не переполняет память)
- Определяет видимость типов. CTS утверждает правила по которым сборки формируют области видимости для типа. CLR дополняет правила видимостью
- Определяет правила, управляющие наследованием типов, виртуальными методами и продолжительностью существования объектов

## 4. Иерархия общей системы типов?

- Тип
  - Типы значения
    - Встроенные типы значения (целые, с плавающей запятой)
    - Перечисления (enum)
    - Структуры
  - Ссылочные типы
    - Самодокументированные типы
      - Встроенные ссылочные типы
      - Классы
      - Массивы
    - Упакованные типы — значения (предназначены для хранения значений типов значений; не объявляются программистом, система автоматически определяет такой тип для любого типа-значения)
  - Указатели
    - Управляемые (представляют собой ссылки на управляемые блоки памяти из кучи среды CLR. В этой куче выполняется автоматическая сборка мусора)
    - Неуправляемые (это обычный указатель на неуправляемый блок памяти из стандартной кучи)
  - Интерфейсы

## 5. Все типы .net делятся на типы значения и ссылочные типы. Определение этих типов?

- Значимые типы — это типы данных, объекты которых представлены фактическими значениями объекта. Если экземпляр типа значения присваивается

переменной, то эта переменная получает новую копию значения. Хранится в ячейках.

- Ссылочные типы — это типы данных, объекты которых представлены ссылкой на фактическое значение объекта. Если экземпляр ссылочного типа присваивается переменной, то эта переменная будет ссылаться на исходное значение. Копирование не происходит (копирование объекта). Описывают так называемые объектные ссылки и представляют адреса объектов. Объекты в куче, ссылки в ячейках.

#### 6. Определение самодокументированных типов

Это ссылочные типы представляющие собой объекты. Каждый объект в куче содержит информацию о своем типе. Основу таким типам составляют классы. Классы могут агрегировать значения других типов, а так же наследоваться друг от друга.

#### 7. Основные характеристики типа классов

<u>Характеристика классов</u>	<u>Описание</u>
Запечатанные	Запечатанные (sealed), или герметизированные, классы не могут выступать в роли базовых для других классов, т.е. не допускают наследования
Реализующие интерфейсы	Интерфейсом (interface) называется коллекция абстрактных членов, которые обеспечивают возможность взаимодействия между объектом и пользователем этого объекта. CTS позволяет реализовать в классе любое количество интерфейсов
Абстрактные или конкретные	Экземпляры абстрактных (abstract) классов не могут создаваться напрямую, и предназначены для определения общих аспектов поведения для производных типов. Экземпляры же конкретных (concrete) классов могут создаваться напрямую
Степень видимости	Каждый класс должен конфигурироваться с атрибутом видимости (visibility). По сути, этот атрибут указывает, должен ли класс быть доступным для использования внешним сборкам или только изнутри определяющей сборки

#### 8. Для чего нужны встроенные типы данных CTS?

В CTS содержится четко определенный набор фундаментальных типов данных. В разных ЯП для объявления того или иного типа используется свое ключевое слово, но в итоге все эти ключевые слова соответствуют одному и тому же типу в сборке mscorlib.dll

#### 9. Если мы хотим создать приложение нейтральное к языкам, что необходимо сделать?

Нужно создать шаблон blank solution. Это позволит в одном приложении создавать несколько проектов, а эти проекты могут состоять из различных ЯП.

#### 10. Прежде чем использовать сборку написанном на одном ЯП, в другом ЯП, что надо сделать?

Компилятор первоначально не знает (не находит), где хранится данный тип который необходимо использовать, не смотря на то, что проекты находятся в одном решении. Для решения проблемы необходимо компилятору сообщить, где находится сборка определяющий данный тип. После этого проект будет присоединен к решению и виден в другом проекте.

**1. Что такое CLI?**

Это спецификация общей языковой инфраструктуры.

**2. Помимо .net, какие еще существуют CLI?**

Есть по крайней мере еще три альтернативы .net framework:

- Mono (linux, mac, windows)
- Rotor (freeBSD, windows)
- Portable.NET (linux, freeBSD)

**3. В чем смысл упаковки?**

Смысл заключается в том, что в куче создается пустой объект нужного размера, а затем значение копируется внутрь этого объекта. Следовательно с помощью упаковки мы можем превратить значение любого типа значения (встроенного, структуры, перечисления) в объект и в дальнейшем работать с этим значением как с объектом.