**Отчёт по практике 14.**

**Студент: Гуртякова А.П.**

**Задача:**Ознакомиться с видео лекцией и ответить на вопросы

**Цель задания:**Написать уникальность понятия Events

**Ответить на вопросы:**

Перечислите все системные типы в пространстве имён System

Чем отличаются делегаты от событий

**Ответ:**

Action<T1,T2…>

Инкапсулирует метод, который принимает до 16 параметров и не возвращает значения.

AssemblyLoadEventHandler

Представляет метод, обрабатывающий событие AssemblyLoad класса AppDomain.

AsyncCallback

Ссылается на метод, который должен вызываться при завершении соответствующей асинхронной операции.

Comparison<T>

Представляет метод, сравнивающий два объекта одного типа.

ConsoleCancelEventHandler

Представляет метод, обрабатывающий событие CancelKeyPress объекта Console.

Converter<TInput,TOutput>

Представляет метод, преобразующий объект от одного типа к другому.

EventHandler

Представляет метод, обрабатывающий событие, не имеющее данных.

EventHandler<TEventArgs>

Представляет метод, обрабатывающий событие, когда событие предоставляет данные.

Func<T1,T2,…,TResult>

Инкапсулирует метод, который принимает до 16 параметров и возвращает значение типа, указанного в параметре TResult.

Predicate<T>

Представляет метод, определяющий набор критериев и соответствие указанного объекта этим критериям.

ResolveEventHandler

Представляет метод, обрабатывающий событие TypeResolve, ResourceResolve или AssemblyResolve объекта AppDomain.

UnhandledExceptionEventHandler

Представляет метод, который будет обрабатывать событие, вызванное исключением, которое не обрабатывается доменом приложения. Сатьюцдлцыв сч

Главное отличие event от delegate состоит в том, что **event может быть запущен только в классе, в котором объявлен**. Помимо этого, при наличии event компилятор создает не только соответствующее приватное поле-делегат, но еще и два открытых метода для подписки и ее отмены на события.

| **Делегат** | **Событие** |
| --- | --- |
| Делегат объявляется с помощью ключевого слова delegate. | Событие объявляется с помощью ключевого слова event. |
| Делегат — это указатель на функцию. Он содержит ссылку на один или несколько методов во время выполнения. | Событие — это механизм уведомления,  который зависит от делегатов. |
| Делегат независим и не зависит от событий. | Событие зависит от делегата и не может  быть создано без делегатов. Событие — это  оболочка вокруг экземпляра делегата, которая  не позволяет пользователям делегата сбрасывать  делегат и его список вызовов и позволяет только  добавлять или удалять цели из списка вызовов. |
| Делегат включает методы Combine() и Remove() для добавления методов в список вызовов. | Класс EventInfo проверяет события и подключает  обработчики событий, которые включают методы  AddEventHandler() и RemoveEventHandler() для  добавления и удаления методов в список вызовов соответственно. |
| Делегат может быть передан как параметр метода. | Событие возникает, но не может быть передано  в качестве параметра метода. |
| Оператор = используется для назначения одного метода, а оператор += используется для назначения нескольких методов делегату. | Оператор = нельзя использовать с событиями, и  только операторы += и -= можно использовать с  событием, которое добавляет или удаляет  обработчик события. Эти методы внутренне  вызывают методы AddEventHandler и RemoveEventHandler. |