1. Что такое делегат? Как определить делегат?

Делегат представляет собой тип , который представляет ссылки на методы с определенным списком параметров и типом возвращаемого значения. Когда вы создаете экземпляр делегата, вы можете связать его экземпляр с любым методом с совместимой сигнатурой и типом возвращаемого значения. Вы можете вызвать (или вызвать) метод через экземпляр делегата.

Делегаты используются для передачи методов в качестве аргументов другим методам. Обработчики событий - это не что иное, как методы, которые вызываются через делегатов. Вы создаете собственный метод, и такой класс, как элемент управления Windows, может вызывать ваш метод при возникновении определенного события. В следующем примере показано объявление делегата:

public delegate int PerformCalculation(int x, int y);

Делегату может быть назначен любой метод из любого доступного класса или структуры, который соответствует типу делегата. Метод может быть статическим или экземплярным. Эта гибкость означает, что вы можете программно изменять вызовы методов или вставлять новый код в существующие классы.

2. Назначение делегатов.

Выше

3. Какие есть способы присваивания делегату адреса метода?

Делегату может быть назначен любой метод из любого доступного класса или структуры, который соответствует типу делегата. Метод может быть статическим или экземплярным. Эта гибкость означает, что вы можете программно изменять вызовы методов или вставлять новый код в существующие классы.

<https://metanit.com/sharp/tutorial/3.13.php>

либо

Message message;

message = Hello;

Либо

message = new Message(Hello);

4. Поясните назначение метода Invoke

Вызов делегата message.Invoke()

Либо massage();

5. Что такое групповая адресация делегата?

Одним из самых примечательных свойств делегата является поддержка групповой адресации. Попросту говоря, групповая адресация — это возможность создать список, или цепочку вызовов, для методов, которые вызываются автоматически при обращении к делегату. Создать такую цепочку нетрудно. Для этого достаточно получить экземпляр делегата, а затем добавить методы в цепочку с помощью оператора + или +=. Для удаления метода из цепочки служит оператор - или -=. Если делегат возвращает значение, то им становится значение, возвращаемое последним методом в списке вызовов. Поэтому делегат, в котором используется групповая адресация, обычно имеет возвращаемый тип void.

6. Как создать событие?

События представляют собой специальный вид многоадресного делегата, который можно вызвать только из класса или структуры, в которых он объявлен (класс Publisher). Если другие классы или структуры подписываются на событие, их методы обработчиков событий будут вызываться, когда класс Publisher будет вызывать событие.

7. Как события связаны с делегатами? Опишите и поясните схему

взаимодействия.

<http://www.electronick.org.ua/articles/sobytiya_i_delegaty_v_yazyke_csharp/>

8. Что такое лямбда-выражения? Приведите пример лямбда-выражения

с несколькими параметрами.Лямбда-выражения представляют упрощенную запись анонимных методов. Лямбда-выражения позволяют создать емкие лаконичные методы, которые могут возвращать некоторое значение и которые можно передать в качестве параметров в другие методы.

9. Что такое ковариантность и контравариантность делегатов? Что это

дает?

<https://metanit.com/sharp/tutorial/3.28.php>

10.Поясните разницу между встроенными делегатами Action и Func.

Делегат Func предназначен для инкапсуляции метода, который принимает в качестве параметров от нуля до четырех аргументов и возвращает значение. Что же касается делегата Action, то единственное отличие его от Func – это то, что Action возвращает процедуру.