1. Что такое делегат? Как определить делегат?

Делегат — это тип, который представляет ссылки на методы с определенным списком параметров и типом возвращаемого значения.

1. Назначение делегатов.

Делегаты используются для передачи методов в качестве аргументов другим методам.

1. Какие есть способы присваивания делегату адреса метода?

Делегату можно назначить любой метод из любого доступного класса или структуры, соответствующий типу делегата. Метод может быть либо статическим, либо методом экземпляра.

С использованием имени метода

С использованием ключевого слова delegate

С использованием лямбда-выражения

1. Поясните назначение метода Invoke.

Метод Invoke в делегатах используется для явного вызова метода, на который ссылается делегат.

1. Что такое групповая адресация делегата?

Групповая адресация — это возможность создать список, или цепочку вызовов, для методов, которые вызываются автоматически при обращении к делегату.

1. Как создать событие?

События объявляются в классе с помощью ключевого слова event, после которого указывается тип делегата, который представляет событие

public delegate void Fine(User obj, int position);

public delegate void Increase(User obj, int length);

public event Fine FineEvent;

public event Increase IncreaseEvent;

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

взаимодействия.

События в C# связаны с делегатами, так как делегаты определяют сигнатуру методов, которые могут быть связаны с событием.

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

с несколькими параметрами.

Лямбда-выражения представляют упрощенную запись анонимных методов.

(string name, int age, string city) => $"My name is {name}, I'm {age} years old, living in {city}."

9. Что такое ковариантность и контравариантность делегатов? Что это

дает?

**Ковариантность: позволяет использовать более конкретный тип, чем заданный изначально** Ковариантность позволяет присваивать делегатам методы, которые имеют более производные (более специфические) типы параметров. То есть, если делегат ожидает метод с базовым типом параметра, то можно присвоить ему метод с производным типом параметра.

**Контравариантность: позволяет использовать более универсальный тип, чем заданный изначально.** Контравариантность позволяет присваивать делегатам методы, которые имеют более общие (более базовые) типы параметров. То есть, если делегат ожидает метод с производным типом параметра, то можно присвоить ему метод с базовым типом параметра.

10.Поясните разницу между встроенными делегатами Action и Func

Делегат Action представляет некоторое действие, которое ничего не возвращает, то есть в качестве возвращаемого типа имеет тип void

Func возвращает результат действия и может принимать параметры. Он также имеет различные формы: от Func<out T>(), где T - тип возвращаемого значения, до Func<in T1, in T2,...in T16, out TResult>(), то есть может принимать до 16 параметров.