1. **Что такое делегат? Как определить делегат?**
   * Делегат — это тип, представляющий ссылку на метод. Делегаты позволяют вызывать методы через их адреса.
   * Определение делегата:

csharp

Копировать код

public delegate void MyDelegate(string message);

1. **Назначение делегатов:**  
   Делегаты используются для:
   * Передачи методов в качестве параметров другим методам.
   * Уведомлений (вместе с событиями).
   * Организации обратных вызовов (callback).
   * Компоновки методов в группу.
2. **Способы присваивания делегату адреса метода:**
   * Использование имени метода:

csharp

Копировать код

MyDelegate del = SomeMethod;

* + Использование анонимного метода:

csharp

Копировать код

MyDelegate del = delegate(string msg) { Console.WriteLine(msg); };

* + Использование лямбда-выражения:

csharp

Копировать код

MyDelegate del = msg => Console.WriteLine(msg);

1. **Назначение метода Invoke:**  
   Метод Invoke вызывает метод, на который указывает делегат:

csharp

Копировать код

del.Invoke("Hello"); // Аналогично del("Hello");

1. **Что такое групповая адресация делегата?**  
   Групповая адресация позволяет связать делегат с несколькими методами. При вызове делегата будут вызваны все методы, связанные с ним:

csharp

Копировать код

del += Method1;

del += Method2;

del();

1. **Как создать событие?**  
   Событие создается через ключевое слово event, связанное с делегатом:

csharp

Копировать код

public delegate void Notify();

public event Notify OnNotify;

1. **Как события связаны с делегатами? Опишите схему взаимодействия:**  
   События — это специализированные делегаты. События позволяют подписывать методы-обработчики, которые будут вызваны при наступлении события. Схема:
   * **Определение делегата:** Указывает на тип методов, которые могут быть подписаны.
   * **Создание события:** Объявляется в классе.
   * **Подписка на событие:** Методы привязываются через +=.
   * **Вызов события:** Через метод Invoke.
2. **Что такое лямбда-выражения? Приведите пример с несколькими параметрами:**  
   Лямбда-выражения — это анонимные методы, упрощенные для краткости.  
   Пример:

csharp

Копировать код

Func<int, int, int> sum = (a, b) => a + b;

Console.WriteLine(sum(3, 5)); // Вывод: 8

1. **Ковариантность и контравариантность делегатов:**
   * **Ковариантность:** Позволяет использовать возвращаемый тип, производный от типа, указанного в делегате.
   * **Контравариантность:** Позволяет использовать параметры метода, более общего типа, чем в делегате.  
     Пример:

csharp

Копировать код

Func<object> covariant = () => "Hello"; // Ковариантность

Action<string> contravariant = obj => Console.WriteLine(obj); // Контравариантность

1. **Разница между Action и Func:**
   * Action: Представляет метод, который ничего не возвращает.

csharp

Копировать код

Action<string> print = msg => Console.WriteLine(msg);

print("Hello");

* + Func: Представляет метод, который возвращает значение. Последний параметр — возвращаемый тип.

csharp

Копировать код

Func<int, int, int> sum = (a, b) => a + b;

Console.WriteLine(sum(3, 5)); // 8