1. Что такое делегат? Как определить делегат?

Делегаты представляют собой объекты, которые указывают на методы. Определение делегата идет с помощью ключевого слова delegate + тип\_вовзращаемого\_значения + название\_делегата

2. Назначение делегатов.

Делегаты позволяют делегировать выполнение некоторому коду извне.

3. Какие есть способы присваивания делегату адреса метода?

Напрямую или через создание объекта делегата через конструктор, в который передается нужный метод.

4. Поясните назначение метода Invoke.

Проверяет передаваемые параметры на пустое значение. Если они пустые, то при вызове такого делегата мы получим исключение.

5. Что такое групповая адресация делегата?

Когда мы создаем один делегат и объединяем в нем два другие – это значит, что в список методов первого делегата войдут методы двух других.

6. Как создать событие?

События сигнализируют системе, что произошло какое-то определенное действие. Чтобы создать событие нужно прописать ключевое слово event, после которого указать тип делегата, который представляет это событие.

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

взаимодействия.

Вначале определяется делегат, который принимает какой-либо параметр, затем с помощью ключевого слова определяется событие, представляющее этот делегат. Определив событие, его можно потом вызвать как обычный метод.

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

с несколькими параметрами.

Лямбда-выражения представляют упрощенную запись анонимных методов. Прописываются как (список\_параметров) => { кусок\_кода\_где\_они\_используются}

9. Что такое ковариантность и контравариантность делегатов? Что это

дает?

Ковариантность делегата предполагает, что возвращаемым типом может быть более производный тип. Контрвариантность – тип параметра может быть более универсальный тип.

10. Поясните разницу между встроенными делегатами Action и Func.

Делегат Action не возвращает значение, Function возвращает.