**Делегат** (delegate) — безопасный указатель на метод. В классе-делегате строго указаны тип возвращаемого значения и аргументы метода (сигнатура).

Все делегаты, являются производными от абстрактного класса **System.MulticastDelegate**, который в свою очередь наследуется от абстрактного класса Delegate.

Экземпляр делегата может ссылаться на любой статический метод или метод экземпляра – при условии, что сигнатура метода совпадает с сигнатурой делегата.

**Анонимная функция** – это оператор или выражение "inline", которое можно использовать каждый раз, когда ожидается тип делегата. Ее можно использовать для инициализации именованного делегата или подставить вместо типа именованного делегата в качестве параметра метода.

Существует два типа анонимных функций, - это Лямбда-выражения и Анонимные методы.

**Лямбда-выражение** — это анонимная функция, которая содержит выражения и операторы и может использоваться для создания делегатов.

**Следующие правила применимы к области действия переменной в лямбда-выражениях.**

- Захваченная переменная не будет уничтожена сборщиком мусора до тех пор, пока делегат, который на нее ссылается, не выйдет за границы области.

- Переменная, введенная в лямбда-выражение, невидима во внешнем методе.

- Лямбда-выражение не может непосредственно захватывать параметры ref или out из включающего их метода.

- Лямбда-выражение не может содержать оператор goto, оператор break или оператор continue, для которых, метка перехода находится вне тела либо в теле содержащейся анонимной функции.

Во всех лямбда-выражениях используется лямбда-оператор =>, который читается как "переходит в". Левая часть лямбда-оператора определяет параметры ввода (если таковые имеются), а правая часть содержит выражение или блок оператора. Лямбда-выражение x => x \* x читается как "x переходит в x, x раз".

Оператор => имеет тот же приоритет, что и оператор присваивания (=) и является право ассоциативным.