**Делегати**

Делегатите представляват .NET типове, които описват сигнатурата на даден метод (броя, типа и последователността на параметрите му) и връщания от него тип. Делегатите приличат на указателите към функции в C и C++ – съдържат силно-типизиран указател (референция) към метод. Те са структури от данни, които приемат като стойност методи, отговарящи на описаната от делегата сигнатура. Чрез тях се осъществяват "обратни извиквания" (callbacks).  
Могат да сочат както към статични методи, така и към методи на инстанция.  
Делегатите в .NET Framework са специални класове и могат да бъдат два вида:  
**-Единични (single-cast) делегати**: Съдържат референция към един единствен метод наследяват класа System.Delegate  
**-Множествени (multicast) делегати:** Съдържат свързан списък от референции към методи наследяват класа System.MulticastDelegate.  
В C# могат да се декларират само Multicast делегати (чрез запазената дума delegate).  
**Multicast делегати** При извикване на multicast делегат, се изпълняват последователно един след друг всички методи от неговия списък. Ако multicast делегат връща стойност или променя ref или out параметър, резултатът е само от последния извикан метод от списъка с методи на делегата. Ако при извикване на multicast делегат някои от методите в неговия списък хвърли изключение, следващите методи от списъка не се извикват. На практика single-cast делегати почти не се използват и под делегат обикновено се има предвид multicast делегат.  
**Класът System.MulticastDelegate** Наследник на System.Delegate и е базов клас за всички делегати в C#. Съдържа метод Combine за сливане на списъците от методи на няколко делегата от еднакъв тип, метод Remove за премахване на метод от списъка за извикване, метод GetInvocationList(), който връща масив от делегати – по един за всеки от методите в списъка за извикване на делегата. Има свойство Method, което описва сигнатурата на методите в делегата.

**Събития (events)** В компонентно-ориентираното програмиране компонентите изпращат събития (events) към своя притежател за да го уведомят при настъпване на интересна за него ситуация. Обектът, който предизвиква дадено събитие, се нарича изпращач на събития (event sender). Обектът, който получава дадено събитие, се нарича получател на събитието (event receiver). За да получават дадено събитие получателите му предварително се абонират за него (subscribe for event).  
 В компонентния модел на .NET Framework абонирането, изпращането и получаването на събитията се поддържа чрез делегати и събития. Събитията в C# са специални инстанции на делегати, декларирани с ключовата дума event. За променливите от тип събитие C# компилаторът автоматично дефинира операторите **+=** и **-=** съответно за абониране за събитието и за премахване на абонамент. Събитията могат да предефинират кода за абониране и премахваме на абонамент.  
**Разлика между събитие и делегат** Събитията, декларирани с ключовата дума event не са еквивалентни на член-променливите от тип делегат. ***public MyDelegate m; ≠ public event MyDelegate m;***Събитията могат да бъдат членове на интерфейс, а делегатите не могат. Извикването на събитие може да става само от класа, в който е дефинирано. Достъпът до събитията по подразбиране е синхронизиран.

**Конвенция за събитията** В .NET Framework се използва утвърдена конвенция за събитията:  
Делегатите, които се използват за събития:  
-имат имена образувани от глагол + EventHandler (SomeVerbEventHandler)  
-връщат void и приемат два параметъра – обект-изпращач от тип System.Object и обект, съдържащ данни за събитието от тип, наследник на System.EventArgs  
Пример: ***public delegate ItemChangedEventHandler( object aSender, ItemChangedEventArgs aEventArgs);***-Събитията се обявяват като public, започват с главна буква и завършват с глагол, например: ***public event ItemChangedEventHandler ItemChanged;***-За предизвикване на събитие се създава protected void метод с име в стил OnVerb, например:  
***protected void OnItemChanged() { … }***-Методът-получател (обработчик) на събитието има име Обект\_Събитие: ***private void OrderList\_ItemChanged() { … }*Събития и интерфейси**Събитията (events) могат да бъдат членове на интерфейси:   
***public interface IClickable   
{  
 event ClickEventHandler Click;  
}***При имплементацията на събитие от интерфейс за него могат да се реализират специфични add и remove методи. За разлика от свойствата, при събитията имплементацията на методите add и remove не е задължителна.  
**Интерфейси, събития, делегати**В .NET поведението "обратно извикване" (callback) може да се реализира чрез интерфейси, делегати или събития  
**Кога да използваме интерфейси?**  
Когато даден обект предоставя съвкупност от много callback методи.  
**Кога да използваме събития?**  
Когато разработваме компоненти, които трябва да известяват своя притежател за нещо, когато търсим съвместимост с компонентния модел на .NET.  
**Кога да използваме делегати?**  
Когато имаме единичен callback метод, който не е свързан с компонентния модел на .NET.