Комуникация между обекти

**1.Събитийното програмиране:**

При събитийното програмиране програмата е в изчакващо състояние и настъпването на определени събития определя кога и коя част от програмния код ще бъде изпълнена. При този вид програмиране изпълнението на програмата зависи от събитията, които ще възникнат (натискане на бутон, минаване с мишката през определена област и тн). Реакцията на определено събитие чрез изпълнение на програмен код не е задължителна. Реакция на събитие, за което липсва програмен код, е стандартна спрямо общоприетите разбирания за ОС (т.е. нищо не се случва).

Събитията могат да бъдат класифицирани в три основни категории:

**·**  Събития, свързани с клавиатурата – настъпват, когато потребителят натисне определен клавиш, например Tab или дори комбинация от клавиши: Ctrl + P.

**·**  Събития свързани с мишката - настъпват, когато потребителят премества мишката, натиска веднъж или двукратно бутон на мишката или влачи показалеца на мишката по екрана.

**·**  Програмни събития – настъпват, когато програмата на Visual Basic направи нещо сама, като например зареждане, отваряне или затваряне на Форма (без да има пряка намеса в този момент от потребителя) или се появи грешка.

Докато в отговор на дадено събитие се изпълнява програмен код за реакцията му може да възникне друго. Програмните участъци, отговорни за реакцията на събития се изпълняват паралелно. Някои събития са взаимосвързани, т.е. едното не може да възникне без преди това да се е случило другото (двукратно щракване на мишката се предшества от еднократно).

При традиционното или "процедурно" програмиране, програмата се разделя на процедури и данни. Една процедура се дефинира от няколко оператора само веднъж и се запаметява. След това може да се ползва на много места в програмата. При процедурното програмиране програмата сама управлява кои части от кода да изпълни и в каква последователност, в зависимост от входните данни.

За разлика от това програмиране, Visual Basic е основан на събитийното програмиране. Събитията могат да настъпят в резултат на действия на потребителя (щракване с

мишката или натискане на клавиш), изпълнение на програмна инструкция или действия на друг прозорец.

При събитийното програмиране, не се следва предварително установен път, а се стартират различни части от кода, в зависимост от настъпилите събития.

**2.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.

**-Видове делегати**

* **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

**3.Събития:**

Събитията позволяват на клас или обект да уведоми други класове или обекти, когато се случи нещо интересно. Класът, който изпраща (или повдига ) събитието, се нарича издател , а класовете, които получават (или обработват ) събитието, се наричат абонати .

В типично C# Windows Forms или уеб приложение вие ​​се абонирате за събития, генерирани от контроли като бутони и списъчни полета. Можете да използвате интегрираната среда за разработка (IDE) на Visual C#, за да преглеждате събитията, които дадена контрола публикува, и да избирате тези, които искате да обработите. IDE предоставя лесен начин за автоматично добавяне на празен метод за обработка на събития и кода за абониране за събитието.

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.