

14-ші дәріс. Делегаттар

- делегаттың анықтамасы;
- делегаттарды қолдану;
- кері шақыру (callback);
- делегатты параметрлер тізімі арқылы беру;
- операциялар.

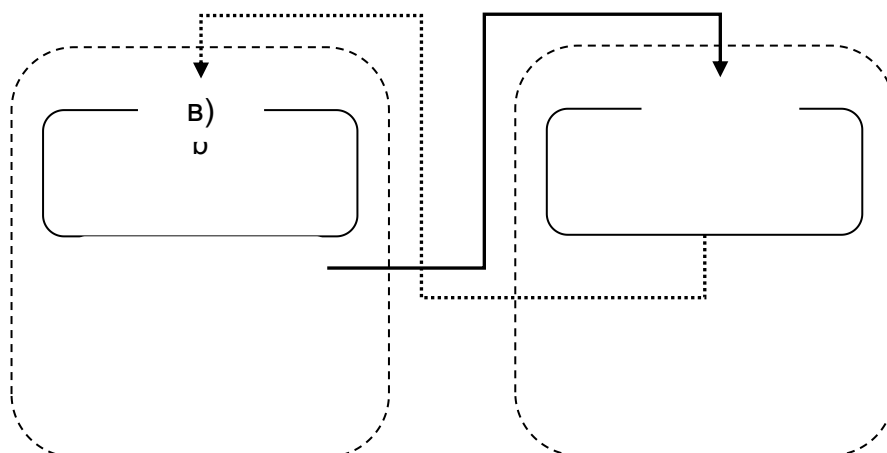
Делегаттың анықтамасы

- *Делегат* — бұл тәсілдерге сілтемелерді сақтауға арналған класс түрі. Делегатты кез кезген класс тәрізді параметр ретінде беруге, осыдан кейін ондағы инкапсуляцияланған тәсілді шақыруға болады.
- Делегаттар оқиғаларды қолдау үшін, сонымен қатар, тілдің тәуелсіз конструкциясы ретінде қолданылады.
- Делегаттың сипаттамасы оның көмегімен шақырылуы мүмкін тәсілдер сигнатурасын анықтайды:
 - [атрибуттар] [спецификаторлар] delegate тип аты(
 - [параметрлер])
 - Делегатты сипаттау мысалы:
 - `public delegate void D ( int i );`
 - Делегаттың базалық класы `System.Delegate` болып табылады

Делегаттарды қолдану

- Делегаттар, негізінен, келесідей жағдайлар үшін қолданылады:
- шақырылған тәсілді компиляция кезінде емес, программаның орындалуы кезінде динамикалық түрде анықтауға мүмкіндік алу;
- объектілер арасында «шығу көзі — бақылаушы» типі бойынша байланысты қамтамасыз ету;
- оларға басқа тәсілдерді беруге болатындай әмбебапты (универсалды) тәсілдерді құру (кері шақырулар механизмін қолдау).

Кері шақыру (callback)



Делегатты параметрлер тізімі арқылы беру

```
namespace ConsoleApplication1 {
    public delegate double Fun( double x );           // делегатты жариялау
    class Class1 {
        public static void Table( Fun F, double x, double b )
        { Console.WriteLine( " ----- X ----- Y -----" );
          while (x <= b)
          { Console.WriteLine( "| {0,8} | {1,8} |", x, F(x));
            x += 1; }
        }
        public static double Simple( double x ) { return 1; }
        static void Main()
        { Table( Simple, 0, 3 );
          Table( Math.Sin, -2, 2 );                      // new
          Fun(Math.Sin)
            Table( delegate (double x) { return 1; }, 0, 3 );
        }
    }
}
```

Операциялар

- Делегаттарды *теңдікпен* немесе *теңсіздікпен* салыстыруға болады. Егер екі делегаттың екеуінде де тәсілдерге сілтеме болмаса немесе екеуінде де бірдей тәсілдерге бірдей реттілікпен сілтемелер жасалған болса, олар тең болады.
- Бірдей типті делегаттармен қарапайым және күрделі *меншіктеу* операцияларын орындауға болады.
- string тіркестері сияқты, делегат та өзгермейтін мәлімет типі болып табылады, сондықтан кез келген өзгерістер жағдайында жаңа экземпляр құрылады, ал ескісін кейіннен қоқыс жинаушы (сборщик мусора) өшіріп тастайды, яғни жояды.
- Делегатты қолдану синтаксисі тәсілді шақыру синтаксисімен бірдей болып келеді. Егер делегатта бірнеше тәсілге сілтемелер сақталатын болса, олар тізбектеле, делегатқа қосылған реттілігімен шақырылады.

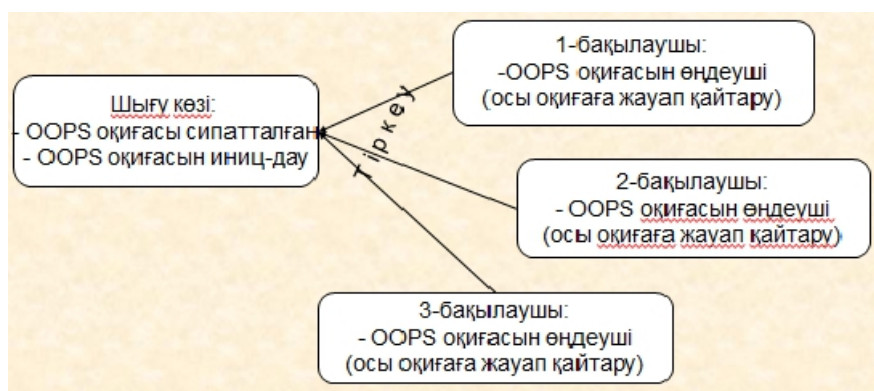
Оқиғалар

- Оқиғаның анықтамасы
- Оқиғалар механизмі
- Делегаттар мен оқиғалар туралы қосымша

Оқиғаның анықтамасы

- *Оқиға* — кластың басқа объектілеріне (бақылаушыларға) өздерінің күйінің өзгеруі туралы хабарламалар жіберуіне мүмкіндік беретін класс элементі.
- Бақылаушы болу үшін объектінің оқиғаларды өңдеушісі болуы керек және ол оқиғалар өңдеушісін шығу көзі объектісінде (в

объекте-источнике) тіркеу керек.



Мысал

```

class Subj { // ----- Оқиғаның шығу көзі
(класс-источник)
    public event EventHandlerOops; // Станд . типті оқиғаны сипаттау
    public void CryOops() { // Оқиғаны
тудыратын тәсіл
        Console.WriteLine( "OOPS!" ); if ( Oops != null )
Oops( this, null ); }
    class Obs { // ----- Бақылаушыкласс
-----
        public void OnOops( object sender, EventArgs e ) { // Оқиғаны
өңдеуші
            Console.WriteLine("Оййй!");
        }
    }
    class Class1 {
        static void Main() {
            Subj s = new Subj();
            Obs o1 = new Obs(); Obs o2 = new Obs();
            s.Oops += o1.OnOops; //
өңдеушіні тіркеу
            s.Oops += o2.OnOops; //
өңдеушіні тіркеу
            s.CryOops();
        }
    }
}

```

Оқиғалар механизмі

- Оқиғалар делегаттар негізінде құрастырылған: делегаттардың көмегімен оқиғаларды өңдеуші тәсілдер шақырылады. Сондықтан класта *оқиғаны құру* келесі бөлімдерден тұрады:

- оқиғалар өңдеушілерінің сигнатурасын анықтайтын делегатты сипаттау;

- оқиғаны сипаттау;
- Оқиғаны тудыратын тәсілді (тәсілдерді) сипаттау.
- Оқиғаның синтаксисі:
- [атрибуттар] [спецификаторлар] event тип аты

Мысал

```
public delegate void Del( object o );      // делегатты жариялау
class A
{
    public event Del Oops;                // оқиғаны жариялау
    ...
}
```

• *Оқиғаларды өңдеу* хабарламаны алушы кластарда орындалады. Ол үшін осы кластарда сигнатурасы делегат типіне сәйкес келетін оқиғаларды өңдеуші тәсілдер сипатталады. Хабарлама алғысы келетін әрбір объект (класс емес!) жіберуші объектіде осы тәсілді тіркеуі тиіс.

• Оқиға — бұл программалаушы үшін ыңғайлы абстракция. Шын мәнінде, ол ішінде делегат экземпляры құрылатын жабық статикалық кластан және өңдеушіні осы делегат тізіміне қосуға және одан өшіруге (жоюға) арналған екі тәсілден тұрады.

• Сыртқы код оқиғалармен бір түрде (сипатта) ғана жұмыс істей алады: өңдеушілерді тізімге қосу немесе тізімнен өшіру, себебі кластан тыс тек += және -= операцияларын ғана қолдануға болады. Бұл операциялар нәтижесінің типі — void, яғни ол арифметикалық типтер үшін күрделі меншіктеу операцияларынан өзгеше болып келеді. Өңдеушілер тізіміне басқаша қатынас құру жолы жоқ.

Мысал

```
public delegate void Del();                // делегатты жариялау
class Subj                                // шығу көзі
{
    public event Del Oops;                // оқиғаны жариялау
    public void CryOops()                // оқиғаны тудырушы тәсіл
    { Console.WriteLine( "OOPS!" ); if ( Oops != null ) Oops(); }
}
class ObsA                                // бақылаушы
{
    public void Do();                    // шығу көзінің оқиғасына жауап қайтару
    { Console.WriteLine( "OOPS екенін көріп тұрмын!" ); }
}
class ObsB                                // бақылаушы
{
    public static void See() // шығу көзінің оқиғасына жауап қайтару
    { Console.WriteLine( "OOPS екенін мен де көріп тұрмын!" ); }
}
```

Мысал

```
class Class1
{
    static void Main()
    {
        Subj s  = new Subj();           // шығу көзі класы
        ObsA o1 = new ObsA();           // бақылаушы
        ObsA o2 = new ObsA();           // объектілері
        s.Oops += new Del( o1.Do );     //
        s.Oops += new Del( o2.Do );     //
        s.Oops += new Del( ObsB.See );  // қосу
        s.CryOops();                    // оқиғаны
    }
}
```

объектісі
кластың
оқиғаға
өңдеушілерді
тудыру

Делегаттар мен оқиғалар туралы қосымша мәліметтер

- Делегатты асинхронды түрде шақыруға болады (жеке ағымда), бұл сәтте бастапқы ағымдағы есептеулерді жалғастыра беруге болады.
- Анонимді делегат (бақылаушы класты құрмайтын):
 - s.Oops += delegate (object sender, EventArgs e)
 - { Console.WriteLine("Мен сендермен біргемін!"); };
- Делегаттар мен оқиғалар өзара байланысқан объектілер арасында олардың үйлесімді күйлерін сүйемелдеуге мүмкіндік беретіндей икемді түрде өзара әрекеттесуді қамтамасыз етеді.
 - Оқиғалар көптеген стандартты .NET кластарына қосылған, мысалы, Windows қосымшаларын құруға арналған Windows.Forms атаулар кеңістігінің кластарына қосылған.