

Teoriaa: Omat delegaatit ja eventit

Olemme käyttäneet toistaiseksi kursseilla muuttujia (kenttiä, parametreja ja paikallisia muuttujia) jotka tallettavat joko C# perustyyppettä tai muita olioita.

Perustyytit toimitetaan aliohjelmille kopioina ellei toisin käsketä (käyttäen ref, out, in avainsanoja).

Perustyyppi (built-in type)	Vastaavan olion luokka
bool	System.Boolean
byte	System.Byte
sbyte	System.SByte
char	System.Char
decimal	System.Decimal
double	System.Double
float	System.Single
int	System.Int32
uint	System.UInt32
nint	System.IntPtr
nuint	System.UIntPtr
long	System.Int64
ulong	System.UInt64
short	System.Int16
ushort	System.UInt16

Lisäksi C# perustyyppisiin kuuluvat alla luetellut kolme tyyppiä jotka muista perustyypeistä poiketen välitetään aliohjelmille referenssinä ([reference](#) types). Niitä ei siis kopioida vaan viittaus/referenssi samaan olion välitetään tarvittaessa koko olion kopiointiin sijaan. Kopion voi luoda, mutta se pitää tehdä *eksplisiittisesti* kutsumalla metodia Clone().

Perustyyppi (referenssi)	Vastaavan olion luokka
object	System.Object
string	System.String
dynamic	System.Object

Delegaatit

C# perustyyppien ja muiden olioiden lisäksi on myös olemassa kolmas ryhmä muuttujia jotka voi tallettaa erilaisiin muuttujiin (kentät, parametrit ja paikalliset muuttujat) ja joita voi siis käsitellä erilaisten datarakenteiden osina: *Taulukoissa*

(Array), listoissa (List), pinoissa (Stack), kasoissa (Heap), puissa (Tree), hash-tauluissa (Hash Table) jne...

Tämä kolmas tyyppi on *delegaattit*. Ne eivät talleta dataa kuten perustyyppit, eivätkä olioiden tiloja (ts. olioiden sisäistä dataa) kuten oliot vaan metodeita, funktioita ja aliohjelmia ts. ohjelmakoodia.

Delegaattit ovat siis muuttujia jotka sisältävät ohjelmakoodin (tai tarkemmin viittauksen siihen).

Se, että ohjelmakoodia voi käsitellä kuten muuttujia helpottaa ratkaisevasti monia ohjelmointiongelmia.

Eventit (tapahtumat)

Eventit eli tapahtumat ovat sisäisesti listoja delegaateista joita jokaista kutsutaan, kun eventtiä kutsutaan. Delegaatti voidaan lisätä eventtiin += operaatiolla.

Jos tutkit Form-luokan koodia jonka Visual Studio Form Designer generoi, löydät jokaista kuunneltua eventtiä kohden tälläisen tapahtumakäsittelijän lisäyksen (subscription). Ts. kun määrittelet Form Designerissä uuden eventin jollekin käyttöliittymäkomponentille tuplakkilaamalla eventtityyppiä, Visual Studio lisää koodiin kaksi osaa:

1. Event handlerin (tapahtumakäsittelijän) jota kutsutaan kun event tapahtuu `Form.cs` tiedostoon sekä...
2. em. event handlerin lisäyksen (tilauksen, subscription) itse eventtiin += operaatiolla `Form.Design.cs` tiedostoon.

HUOMAA että jos poistat event handlerin koodin, sinun tulee poistaa myös sen lisäys eventtiin. Ts. joudut poistamaan generoidun koodin molemmista tiedostoista! Jos jätät eventin lisäyksen `Form.Design.cs` tiedostoon, ei Form Designer käynnisty, eikä ohjelma luonnollisesti käänny.

Eventit lisäksi takaavat että muista osista ohjelmaa ei voida käydä poistamassa muiden olioiden tekemiä eventtitilauksia.