# 4. előadás / gyakorlat

## Végigvettünk majdnem mindent

Attribútumokra legfeljebb kitérünk

Két hét múlva WP7 fejlesztés. Hello World Windows Phone-ra

Újdonságok:

* Új Office támogatáskönyvtára
* LINQ, Lambda kifejezések, Linq to objects, xml, datasets, SQL.
* Extension Method (Statikus (azaz nem példányosítható) osztályba rakandó függvény, ahol a paraméterre a $this$ kulcsszó hivatkozhatunk, mint az indexer.)
* XAML szerkesztő
* Expression Blend (mára nagyrészt megszűnt)
* Implicit típushasználat: int, string, stb helyett var kulcsszó. Kitalálja, milyen típus lesz. Csak lokális változó lehet implicit.
* Anonymous típus: Nem ismert előre a változó típusa. $var miVagyok = new \{név="Joe", Kor=20\}$
* Automatikus tulajdonság: $public string Név \left\{get;set;\right\}; public string readonlyNév \left\{get;private set;\right\};$
* Objektuminicializálás egyszerűsödött, Auto tulajdonság, Tömbök közvetlen inicializálása közvetlen kezdőértékekkel.
* Lambda kalkulus: Csak függvényt ismer, csak változóbehelyettesítés műveletet ismer. Ebből vettünk át C#-ba.
	+ Lambda operátor a $=>$ Példa lambda kifejezés: $x=>x+5$
	+ Delegált definiálása anonymous lambda funkcióval egyszerűbb.
	+ Van előre definiált egyparaméteres $Func$ delegált. Használata: $Func<int, bool>myFunc=x=>x==5; myFunc\left(4\right)$ hamis
	+ Kikerült a $delegate$ régi módi használata a divatból.
	+ Van $void$os, $Generic$ bemeneti paraméterrel is.
* LINQ adatlekérdező nyelv
	+ Adatforrást kell definiálni, lekérdezést definiálni, majd a végrehajtás.
	+ Adatforrások: Memóriában levő változó ($IEnumerable$, azaz akár sima tömb is vagy $IQueryable$ lekérdezhető elemnek kell lenni), XML (fájl), adatbázis
	+ Kétféle eredmény: egyszerű adat vagy sorozat.
	+ Szintaxisa az SQL fordítottja: $from…\left(where…\right) select$
	+ $from x in nums where \left(x\%2\right)==2 select x;$
	+ vagy két delegálttal: $scores. Where\left(h=>g>80\right).Select\left(s=>s\right)$
	+ A fordító az utóbbira alakítja át az első példát.
	+ Összetett példa:

$$var query= words.GroupBy\left(w=>w.Length,w=>w.ToUpper\left(​\right)\right).Select\left(g=>new\left\{Length=g.Key, Words=g\right\}\right).OrderBy\left(o=>o.Length\right);$$

$$foreach\left(var obj in query\right)\left\{\begin{array}{c}Console.WriteLine\left(​\right);\\foreach\left(string word in obj.Words\left\{Console.WriteLine\left(​\right)\right\}\right)\end{array}\right\}$$

* Láttunk egy példát $async$-ra.
	+ $HTTPClient$ régóta megvan. De csak mostanra nem fagyasztja le a programot. Fut tovább az információ nélkül.
	+ Saját $async$ függvényünk is lehet, visszatérési értéke a $Task<>$
	+ Háttérszálon fut. Ha mégis le akarsz fagyni, ott az $await$ kulcsszó. vagy a $.Result\left(​\right)$ hívás a $Task$ változón.
	+ $void$os $async$ is lehet, ekkor nem generikus $Task$ a visszatérési értéke.
* $dynamic$: Office cellákat nem kell konvertálgatni.
* Nevesített paraméter kettősponttal megadása: Így a sorrend változtatható. Hatékonyságot csak futásidőben veszítünk.
* Kovariáns és kontravariáns
	+ Kovariáns ($covariant$) visszatérési típus utódtípusát is használhatjuk. Módosítószó: $out$
	+ Kontravariáns ($contravariant$) bemenő paraméter őstípusa is lehet bemenet. Módosítószó: $in$)
* Csak interfészben használhatóak.
	+ Object tömbbe pakolhatunk stringeket például.

Ezzel a nyelvi jellemzőket végigvettük.

Jövő héten még C#-os gyakorlat lesz, rutinszerzéshez. ($Generic$ + kivételkezelés + $extension$ metódus + delegált)

WP7 két hét múlva.

Előadás vége.