# Webes és mobil programozás F#-ban 6. gyakorlat

## Berezvai Dániel jegyzete <http://elte.3ice.hu/>

Csak én jöttem be. UPDATE: Ketten jöttek még, később.

## Beadandóm

F# szótárkivétel-szerver, készülékek és operációs rendszerek közötti szinkronizáláshoz.

## OOP példa

Lesson4.md

Rekordként megadott pont2D, erre függvények.

Tag funkciók definiálhatók, például két pont távolsága.

Statikus tag origó.

Eltolás.

 **type** Point2D **=**

 **{** X int**;**

 Y int **}**

 member p**.**DistanceFromPoint**(**otherPoint2D**)** **=** **(**p**.**X**-**other**.**X**)(**p**.**X**-**other**.**X**)+(**p**.**Y**-**other**.**Y**)(**p**.**Y**-**other**.**Y**)** float sqrt

 static member Origo **=** **{**X**=**0**;** Y**=**0**}**

 member p**.**ShiftX**(**x**)** **=** **{**p **with** X **=** p**.**X**+**x **}**

 member p**.**ShiftY**(**y**)** **=** **{**p **with** Y **=** p**.**Y**+**y **};;**

### Példa használat

 **let** point **=** **{**X **=** 5**;** Y **=** 6**}**

 point**.**DistanceFromPoint**({** X **=** 2**;** Y **=** 3 **});;**

 point**.**DistanceFromPoint**(**Point2D**.**Origo**);;**

 printfn **%**A **(**point**.**ShiftX**(**6**));;**

Behúzás kell tagok elé

Használjuk, mert rövidebb és olvashatóbb forráskód lesz.

### Lehetséges kiterjesztés

Legyen a pontból vektor, origótól számított hosszal, amit egyszer kiszámolunk, majd tárolunk.

## Osztály

 **type** Point2D**(**xint**,** yint**)** **=**

 member p**.**X **=** x

 member p**.**Y **=** y

 member p**.**DistanceFromPoint**(**otherPoint2D**)** **=** **(**p**.**X**-**other**.**X**)(**p**.**X**-**other**.**X**)+(**p**.**Y**-**other**.**Y**)(**p**.**Y**-**other**.**Y**)** float sqrt

 static member Origo **=** Point2D**(**x **=** 0**,** y **=** 0**)**

 member p**.**ShiftX**(**dx**)** **=** Point2D**(**x **=** x **+** dx**,** y **=** y**)**

 member p**.**ShiftY**(**dy**)** **=** Point2D**(**x **=** x**,** y **=** y **+** dy**)**

Alapértelmezett (implicit) konstruktor megadva.

Getter metódusok.

Műveletek: Fenti klónozás helyett objektumot hoz létre.

### Példa

 **let** point **=** Point2D**(**3**,**5**)**

 point**.**ShiftX**(**6**).**X printfn **%**A

Nem mindegy, hogy az F# változót vagy értéket vár. Be kell csomagolni objektumnak

### Megjegyzések

Felkiáltójel jelöli az objektum továbbküldést: $|>!$

Operátor és metódusnév túlterhelés is van, mint más nyelvekben megszoktuk.

Öröklődés is van, $inherit$ kulcsszóval.

## Aszinkron programozás

Eddig programjaink mind egy-egy szálon futottak.

$async$ blokk segítségével kérhetjük, hogy külön szálon fusson minden hívás.

### Példa: async map

Lesson5.md → {1}

Felkiáltójelek speciális jelentéssel bírnak:

$let!$ aszinkron workflow

**let!** var **=** expr **in** body

Másik: $return!$

Jelentése: "Értékelt ki az egész aszinkron munkamenetet!"

## Sitelet példa miből áll

Union típus

Modul definíció

Rekord

Operátor definíció

IWebsite interfész

Az $|> Async.start$ utasítás fordul át majd JavaScript AJAX hívásra.

Nem sikerült lefordítani.

Gyakorlat vége.