# <u>Apple Swift kurzus</u> <u>5. gyakorlat</u>

### MVC modell:

*<u>Model</u>:* - a program funkcionalitásának helye (osztályok, attribútumok, metódusok és azok feladatainak definiálása

- pl.: egy számológép alkalmazás esetén a számológép "agya", milyen funkciókat kell megvalósítania egy számológépnek

*<u>View:</u>* - a grafikus felület összeállítása (UI elemek)

- pl.: egy számológép esetében a kijelző, a gombok megjelenése a megfelelő helyre és módon kerüljön

Controller: - a View és a Model összekapcsolása

- a funkcionalitás hozzárendelése a UI elemekhez

- pl.: egy számológép esetén az összeadás gomb lenyomásakor hívódjon meg az összeadás metódus

### Példa alkalmazás ennek használatára (PizzaDemo):

# - Hozzunk létre egy új projektet (iOS → Single View Application)

Ami fogad bennünket:

- ~ AppDelegate.swift: az alkalmazás életciklusát szabályozza
- ~ Assets.xcassets: az alkalmazáshoz hozzárendelt fájlok könyvtára (pl.: képek)
- ~ Info.plist: az alkalmazás információit tartalmazza (pl.: terméknév, verziószámok, stb...)
- ~ ViewController.swift: itt kerül implementálásra az MVC modell Controller része
- ~ Main.storyboard: itt szerkeszthető grafikusan az MVC modell View része

~ A fent említett fájlok egy mappába vannak szervezve. A fordításkor létrejövő .app fájl a Products mappába kerül.

~ A mappákat magába foglalja a projekt.

~ Ha több projekttel szeretnénk egyszerre dolgozni, azokat egy Workspace-be kell foglalni.

# - Hozzunk létre egy új Swift fájlt, és ezt helyezzük el a többi .swift kiterjesztésű fájl mellé. A fájlnak adjuk a Pizza.swift nevet.

~ Ebben a fájlban definiáljuk a Pizza osztályt (**class Pizza {}**), annak összes attribútumát, illetve metódusát.

~ Ez a fájl tartalmazza az MVC Model részét.

#### - A Main.storyboard-on tervezzük meg az alkalmazás megjelenését.

~ Mindenekelőtt győződjünk meg arról, hogy a storyboard-on levő View Controller objektum rendelkezik belépési ponttal (*Is Initial View Controller*), illetve a hozzátartozó osztály a ViewController.

S	
Inferred 📀	
Inferred 📀	
Inferred 📀	
Inferred 📀	0 🗉 🕂 🛙 🕀
	Custom Class
	Class ViewController O
	nferred O nferred O nferred O nferred O

~ Emeljünk be Label-t, amely kiírja az általunk kijelölt Pizza típusát, méretét és területét (a kék segédvonalak segítenek a pontosabb elhelyezésben). A Label-t méretezzük úgy, hogy a benne levő szöveg jól olvasható legyen (*Attributes inspector*).

~ A Label alatt helyezzünk el egy nyomógombot (*Button*), amelynek segítségével a lehetséges pizzaméreteket lehet kiválasztani. Az elkészített nyomógombot sokszorosítsuk (copy/paste).
~ Helyezzünk el ezek alatt egy szegmentált vezérlőt (*Segmented control*), amely a pizzák ízesítését segítenek beállítani. A vezérlő attribútumainál megadható, hogy hány szegmens legyen látható, majd mindnek külön érték is adható.

 $\sim$ Végül egy Cancel feliratú gombot tegyünk az egész alkalmazás legaljára, mely törli az eddigi adatbevitelt.



### - A UI elemekhez készítsünk hivatkozásokat, hogy azokat a kódunkban is elérhessük.

~ A storyboard megnyitását követően jobb felül váltsunk át Assistant editor funkcióra (osztott

képernyő). Ezzel egyszerre látható a UI felület és a programkód. 🔳 🔯 🛁

~ A jobb oldalon látható fájl a ViewController.swift legyen, hiszen az ebben levő osztállyal (ViewController) van összekötve a storyboard.

~ A bal oldali objektumokból készítsünk hivatkozásokat, amiknek segítségével programozhatóak lesznek a UI elemek. Tegyük meg a következőképp: jobb egérgomb az objektumon, majd húzzuk az egeret lenyomva tartott egérgombbal a kódterületre.



~ A felbukkanó ablakban adjuk meg, hogy az adott objektum csak megjelenítést szolgál (Outlet) vagy valamilyen funkciót kell végrehajtania érintéskor (Action). A hivatkozott objektumot nevezzük el, illetve a hivatkozás típusa minden esetben az objektum típusa legyen (ne AnyObject!)

Connection	Outlet
Object	View Controller
Name	myLabel
Туре	UlLabel
Storage	Weak

# - Az egyes UI elemek hivatkozásainak segítségével hozzájuk rendelhetjük a funkcionalitásokat.

- ~ Ezt mind a ViewController osztályon belül tesszük meg.
- $\sim$  a **viewDidLoad()** metódus felelős a kezdeti megjelenésért

- A kész alkalmazást futtassuk a Play gomb megnyomásával.

