

Rétegzett alkalmazás - fejlesztés

• Háromrétegű architektúra

- Adat, logikai, megjelenítési réteg
- Felső és az alsó rétegek között egy közbeeső rétegen keresztül kommunikálnak.
- Logikai réteg
 - Feladata a funkciók, üzleti logika megvalósítása, amit az adat rétegtől kapott adatok alapján feldolgoz.
- Adat réteg
 - Adatok, információk tárolása
 - Például adatbázisból kapott adatok, amit továbbít a logikai réteg felé.
- Megjelenítési réteg
 - Adatok megjelenítése az üzleti logika alapján a felhasználó felé.

• Fontos, hogy betartsuk a követelményeket:

- Karbantarthatóság; újrafelhasználhatóság (pl.: DLL, npm); Szálazhatóság

• MVVM → Model - View - ViewModel

- Általában WPF alkalmazások rétegzett fejlesztésére használjuk.
- Model → Egység-összetétel tartalmazza, ObservableObject
- ViewModel → Itt hívjuk meg a logic-ot és hozzájuk lehet például a parancsok.
 - ICommand → parancsok
 - Observable Recipient
- View → Nézet, XAML fájl
 - DataContext-ben meghívjuk a ViewModel-t



- View tud a ViewModelről \Rightarrow DataContext, adatköltés miatt (Binding)
- Model nem tud a ViewModelről, de a ViewModel tud a Model-ről

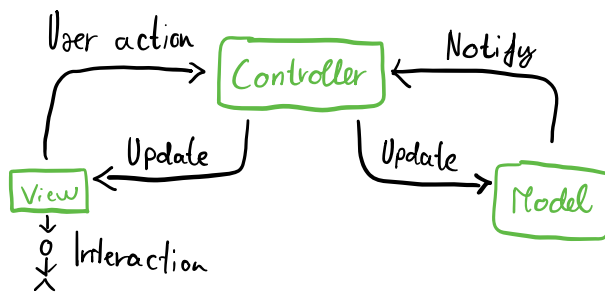
• Binding

- Adatköltés \rightarrow lehet egy vagy több irányú

• MVC

- ASP.Net-ben használjuk a webalkalmazások fejlesztésére.
- Model \rightarrow Egyed osztályoz
- View \rightarrow Nézet, amit a felhasználó lát
- Controller \rightarrow Felhasználói interakciók kezelése

MVC Folyamata



1. Felhasználó csinál vmit a UI-on (View)
2. View tájékoztatja a Controller-t
3. Controller frissíti a Model-t
4. Model értesíti a Controllert
5. Controller frissíti a View-t