

## TEMA 1

Model-view-controller (MVC) este un model arhitectural utilizat în ingineria software. Succesul modelului se datorează izolării logicii de business față de considerentele interfeței cu utilizatorul, rezultând o aplicație unde aspectul vizual sau/și nivelele inferioare ale regulilor de business sunt mai ușor de modificat, fără a afecta alte nivele.

MVC este util atunci când vă planificați aplicația, deoarece vă oferă o schiță a modului în care ideile dvs. ar trebui organizate în codul real. Este o modalitate de a organiza funcțiile de bază ale codului în propriile cutii bine organizate. Acest lucru vă permite să revizuiți aplicația și să vă împărtășiți aplicația cu alții mult mai ușor și mai curat.

Într-o aplicație tipică, sunt trei părți fundamentale :

- ✓ Model – modelează datele și regulile de funcționare (acces și actualizare date)
- ✓ View - prezintă conținutul modelului în modul specificat, accesând datele din model și trebuie să păstreze consistența prezentării datelor atunci când modelul se schimbă
- ✓ Controller - :translatează interacțiunea cu view în acțiuni care să fie efectuate de model.

Interacțiunea înseamnă:

- ♣ Pentru o aplicație desktop apăsări de butoane sau selecții de meniuri.
- ♣ Pentru o aplicație Web cereri HTTP GET sau POST.

Comunicare se realizează după cum urmează:

1. Parte de view primește evenimente de la utilizator pe care le transmite mai departe spre partea de controller
2. Partea de controller primește evenimentele de la utilizator prin partea de view
3. Partea de controller modifică partea de model în raport direct cu răspunsurile de la utilizator (sau, în unele cazuri, partea de controller modifică direct partea de view și numai actualizează partea de model deloc)
4. Partea de model se modifică în urma "update"-urilor de la partea de controller
5. Partea de model anunță părții de view modificările
6. Partea de view actualizează interfața utilizator( de exemplu o anume aranjare sau o retrasare a interfeței, sau se poate chiar prin emiterea unui sunet).

Un exemplu concret care să ne ajute să înțelegem cum funcționează este următorul : Să ne gândim la modul în care preparăm cina. Avem un frigider plin de alimente, care este ca modelul. Frigiderul (Model) conține materiile prime pe care le vom folosi pentru cină. Probabil că avem, o rețetă (presupunem că o urmă exact), reprezentând Controller-ul. Rețetele dictează lucrurile din frigider pe care le scoatem, cum le vom pune împreună și cât timp trebuie să gătim. Avem de asemenea tacâmuri, asemănătoare cu un View. Ele permit invitaților să interacționeze cu Model și Controller.