

## **Temă 1 PSSC- Model-View-Controller**

**Model-view-controller** 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.

**Model:** Această parte a controlatorului manipulează operațiunile logice și de utilizare de informație (trimisă dinainte de către rangul său superior) pentru a rezulta de o formă ușor de înțeles.

**View:** Acestui membru al familiei îi corespunde reprezentarea grafică, sau mai bine zis, exprimarea ultimei forme a datelor: interfață grafică ce interacționează cu utilizatorul final. Rolul său este de a evidenția informația obținută până ce ea ajunge la controlor.

**Controller:** Cu acest element putem controla accesul la aplicația noastră. Pot fi fișiere, scripturi (eng. scripts) sau programe, în general orice tip de informație permisă de interfață. În acest fel putem diversifica conținutul nostru de o formă dinamică și statică, în același timp.

### **Avantaje:**

- **Dezvoltare Simultană:** Mai mulți dezvoltatori pot lucra simultan pe un model, controller sau pe un view.
- **High Cohesion:** MVC permite gruparea logică a acțiunilor conecate pe un controller. View-urile pentru un anumit controller sunt de asemenea grupate împreună.
- **Low Coupling:** În natura MVC există o cuplare scăzută între Models, Views și Controllers.
- **Ușurința Modificării:** Din cauza separării responsabilităților, dezvoltarea sau modificarea viitoare este mai ușoară, adică se mărește scalabilitatea produsului.
- **View-uri Multiple:** Un model poate avea mai multe view-uri.

### **Dezavantaje:**

- **Complexitatea:** O aplicație este mai greu de dezvoltat dacă se folosește Arhitectura MVC.
- **Wide Range App:** MVC nu este făcut pentru aplicații mici. Asupra acestora pot avea “efecte adverse” în ceea ce privește performanță și designul.
- **Isolated Development:** Procesul de dezvoltare pentru autorii UI-ului este limitat.
- **Legături:** În MVC între View și Controller este o legătură foarte strânsă. Acestea sunt separate una de alta, dar oarecum legate prin componente, așa că un view fără controller este foarte limitat și invers, blocându-se astfel utilizarea lor independentă.
- **Accesul la date:** Potrivit unui Model, într-o interfață de operare un View poate necesita mai multe apelări a unei metode pentru a primi suficiente date. Aceste apeluri frecventem “inutile” nu afectează datele, dar afectează foarte mult performanța.