

Design Mönster



VAD OCH VARFÖR OCH TILL VILKEN NYTTA?



Det finns två typer av Design Mönster. *Pattern* och *Anti Pattern*.

De är lösningar till problem som utvecklare har mött och sen löst. Då har man insett vikten av att behålla sin lösning, och skapa ett pattern.

Tanken är man inte ska upprepa sin kod, men ibland till kostnad mot brott mot Javas grundregler.

Abstraction Encapsulation Inheritance Polymorphism

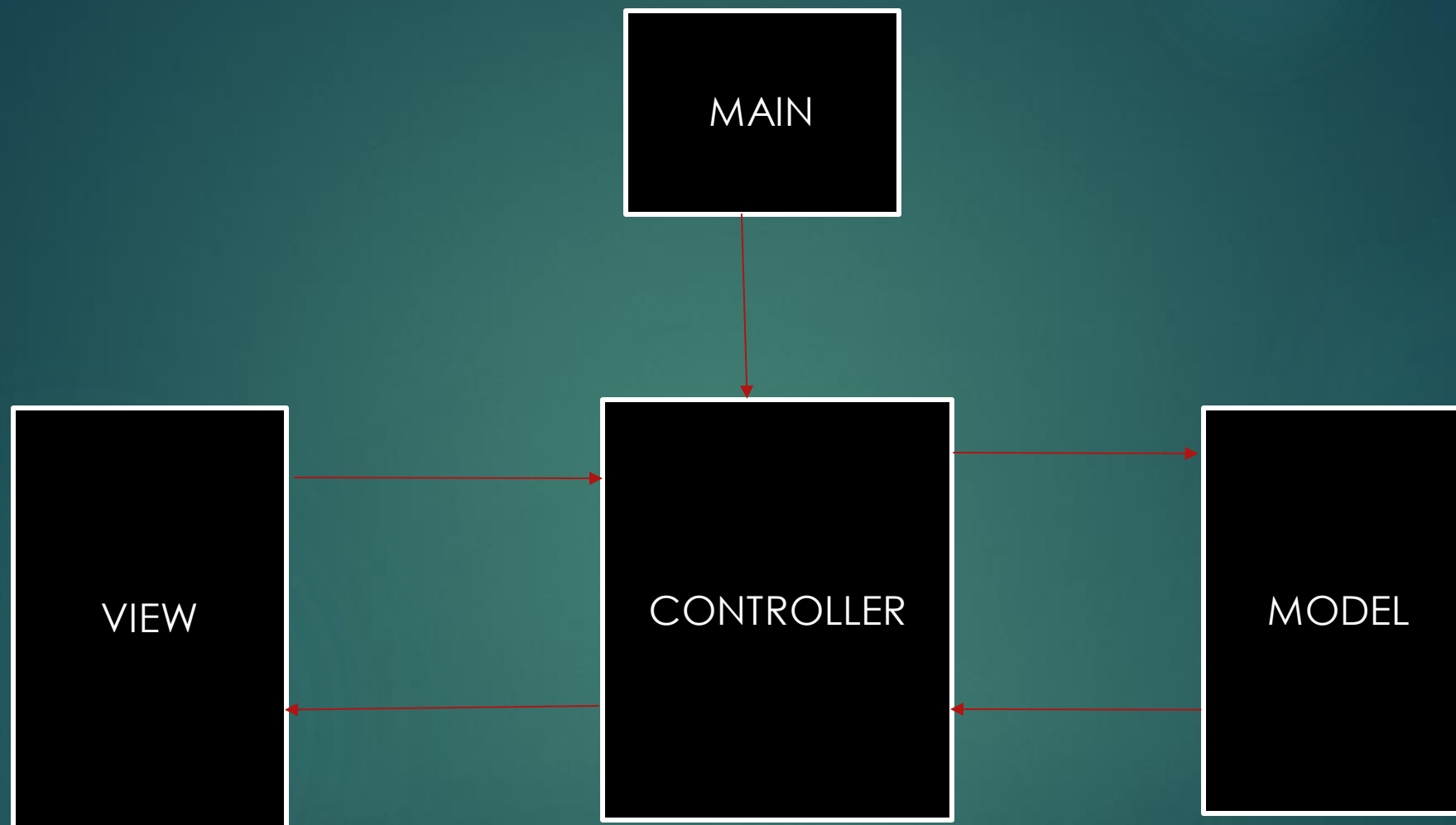
Model – View – Controller



MVC är till för att seperera en applikations uppgifter. Betraktas som ett anti pattern, men används väldigt ofta då nyttan är större än brottet.

Kort sagt *Instansierar* man ett Objekt, som sen anropar *Controllern* som ligger mitt emellan *View* och *Model*. *Model* är ren POJO, där alla beräkningar / transaktioner ska genomföras.

View ska givetvis enbart vara visualiseringen av POJO:n



Observer



Observers används när man vill nå ut till flera klasser samtidigt för att berätta om en *förändring* som man övervakar.

Man börjar med att skapa en *Observer* klass som övervakar en speciell klass. Denna *Observer* klass *måste* vara abstract, och innehålla en abstract metod.

Klassen som ska övervakas måste använda sig av denna abstracta klass med metoderna `setState` och `attach`.



Därefter skall varje klass som ska ha tillgång till Observern ärva den abstracta klassen och sen i konstruktorn ha en parameter med den klassen som ska övervakas.

Sen måste den abstracta metoden implementeras så när den övervakade klassen får en förändring, kommer detta ut till alla klasser som ärvt in Observern.