

# Imhotep

Vidensformidling indenfor  
Softwareudvikling



## Fleksibelt software gennem Design Patterns og Frameworks.

### Overblik:

I kurset vil jeg fokusere på teknikker til at sikre fleksibelt software, forstået som software hvor omkostningerne forbundet med ændrede krav er relativt lave. Kurset vil naturligt fokusere på klassiske design patterns og vil gennemgå en lang række af de klassiske fra den legendariske "Gang of Four" bog. Imidlertid vil kurset lægge stor vægt på de centrale teknikker og den design forståelse som ligger til grund for "magien" i design patterns: en beherskelse af disse teknikker giver bedre software end blot at kunne en stor mængde design patterns. De primære teknikker er *principperne for kompositionelt design*, *rollebegrebet* og *3-1-2 processen* som sikrer at produktvariabilitet designes med lav kobling og høj samhørighed, *refaktorisering* som fokuserer på at sikre stadig høj kvalitet i en software arkitekturen under forandring, og endelig *frameworks* som definerer principperne for produktlinjer og sætter de øvrige begreber ind i en kontekst. Endvidere vil jeg komme naturligt ind omkring den stærkt øgede *testbarhed* som et fleksibelt design kan give, samt *test doubles* som definerer teknikker til at teste software uafhængigt af eksterne ressourcer.

Teknikkerne vil blive direkte anvendt på at udvikle en produktlinje, en betalingsautomat til en parkeringsplads. Dette software starter som en simpel applikation men i præsentationerne gennemgår jeg en proces hvor nye kundekrav og arkitektur-refaktoriseringer afløser hinanden og udvikler produktet til en pålidelig og fleksibel konfigurerbar produktlinje. Denne proces er også udgangspunktet for hand-on øvelserne i kurset. Jeg vil bevidst fokusere på teknikernes kode-nære aspekter fordi alle fordelene i design patterns let kan ødelægges ved simple fejl på implementationsniveauet.

### Format:

Kurset vil skifte mellem præsentationer og dialog med kursisterne, hands-on sessioner hvor der løses design- og programmeringsopgaver i direkte forlængelse af de gennemgåede teknikker i grupper af 2-3 personer, og diskussioner af erfaringerne fra disse øvelser.

### Forudsætninger:

Jeg forudsætter at deltagerne kan læse gængs UML 2.0 notation (specielt klasse diagrammer og sekvens diagrammer). Jeg forudsætter at deltagerne kan programmere i et objekt-orienteret programmeringssprog og kender standard objekt-orienteret terminologi. Opgaverne i kurset vil være i Java. Kursisterne skal have adgang til Windows/Linux baserede PC'er (en per gruppe af 2-3 personer) som er installeret med et

# Imhotep

Vidensformidling indenfor  
Softwareudvikling



Java udviklingsmiljø og editor (jeg vil fremsende en installations ZIP/CD i god tid med den relevante software). Udviklingsmiljøet er et simpelt (Shell baseret) så tiden ikke spildes med at sætte sig ind i et uoverskueligt udviklingsmiljø. Det forudsættes at der maksimalt er 15 kursister på kurset for at sikre at jeg får diskuteret med hver enkelt gruppe under øvelserne.

## Litteratur:

Kurset baserer sig på min bog *Flexible, Reliable Software: Using Patterns and Agile Development*, CRC Press 2010. Jeg forventer at I indkøber et antal eksemplarer som grupperne kan benytte i deres øvelser.

## Rettigheder:

Slides og opgavemateriale må frit bruges i Terma men ikke videregives til tredje-part. Jeg ejer rettighederne til alt materialet og må anvende det i anden sammenhæng.

## Dagsplan (udkast):

Undervisningen foregår to dage i træk, fra kl. 9.00 til 16.00. Der beregnes ca.  $\frac{3}{4}$  time til frokost, samt to pauser af ca. et kvarter i løbet af undervisningsdagen.

### Dag 1:

Introduktion af Case, en betalingsautomat for en parkeringsplads. Håndtering af en ny produktvariant: Analyse af variabilitetsteknikkerne "source code copy", "parametrisk-", "polymorfisk-", og "kompositionelt" design. Udledning af Strategy pattern igennem et kompositionelt design og 3-1-2 processen. Refaktorisering som teknik til redesign på en pålidelig måde. Ny produktvariant og efterfølgende udledning af State pattern. Teknikker til test med kontrol af eksterne resurser: Test Doubles. Ny produktvariant og efterfølgende udledning af Abstract Factory pattern.

### Dag 2:

Pattern skrøbelighed. Rolle begrebet, principperne for kompositionelt design og multi-dimensionel varians. Centrale patterns: facade, decorator, adapter, null object, command, observer, model-view-controller. Frameworks og software produkt linjer.