

ITU-VOOP-F2002

Design Patterns 2

Lars Thorup

Dagens overblik

- Opsamling fra sidst
- Design Patterns
 - Command
 - Strategy
 - Decorator
 - Flyweight
 - Builder
 - Proxy
- Til næste gang

Opsamling fra sidst

- Byggescript
- Testcases
- JUnit
- Effektivitet
- Anden delaflevering: 8. april

Opsamling fra sidst

- Første runde patterns
 - Composite
 - Iterator
 - Visitor
 - Abstract Factory
 - Singleton
 - Observer

Command – 1

- Bruges til: At håndtere kommandoer som objekter for at kunne lave historik, køer, undo/rollback
- Også kaldet: Action, Transaction, Callback.
- Eksempler:
 - UserInterface kommandoer knyttet til forskellige stimuli
 - Kommunikation med tråde via en kø
 - Undo
 - Recovery
 - Makroer

Command – 2

- Pointe: Fælles Command interface til alle kommandoer med operationerne execute og undo
- Issue: Command context: Commands kan instantieres med viden om deres kontekst (`_document`) eller viden om hvordan den skaffes: `Application.getInstance().getCurrentDocument()`
- Issue: Command parametre (fx filnavn): Commands må spørge brugeren eller parametre kan persisteres sammen med Command objektet.

Command – 3

- Issue: Macroer ved at kombinere med Composite pattern, således at MacroCommand indeholder en sekvens af Commands.
- Issue: Undo ved at gemme kopier af Command objekter i en historik liste og sørge for at de rummer information til at kunne undo'e.
- Ved Undo og Redo (herunder Macro-kald) bruger den de argumenter den samlede tidligere.

Flyweight – 1

- Bruges til: At dele objekter med hyppig fælles tilstand for at spare plads.
- Eksempler: State-less objekter i en belastet web-server.
- Pointe: At "skabe" et nyt objekt ved at bede en Flyweight factory om et objekt, der kan være delt.

Flyweight – 2

- Issue: Kontekst kan ikke lægges i objekter men må hver gang sendes med ved kald. Kræver at konteksten er let at få udefra. Giver dårligere performance.
- Issue: Applikationen må ikke have brug for at teste objekt identitet, da denne er kontekstbestemt.

Proxy – 1

- Bruges til: At foregøjle direkte adgang til et objekt man ikke ønsker direkte adgang til.
- Kaldes også: ambassadør, stedfortræder
- Eksempler: Store datastrukturer der ligger i filer/database, søgeresultater, interprocesskommunikation.
- Pointe: Have et fælles interface som både det rigtige objekt og proxyen implementerer. Proxyen har en reference tildet rigtige objekt.

Proxy – 2

- Issue: Vide visse ting om objektet så direkte adgang kan undgås. Copy-on-write.
- Issue: Kunne repræsentere det rigtige objekts identitet selv om det ikke er skabt endnu: filnavn, netnavn, registernøgle.

Decorator – 1

- Bruges til: På runtime at tilføje ekstra funktionalitet til et objekt uden at klienter opdager det.
- Også kaldet: Wrapper.
- Eksempler: Widget pynt, stream egenskaber
- Pointe: Implementerer samme interface som det objekt den dekorerer og refererer desuden til et sådant objekt.

Decorator – 2

- Issue: Når en række egenskaber ved et objekt skal kunne til- og fravælges i vilkårlige kombinationer.

Strategy – 1

- Bruges til: At vælge blandt et sæt af algoritmer på run-time.
- Også kaldet: Policy
- Eksempler: Echo cancellation, UtteranceStarted detection, Orddeling, Layout Management, Register allokering i en compiler, Robust eller fast iterators. Print Layout. Input Validator.
- Pointe: Istedet for at if'e på flag rundt omkring i koden, så indkapsl den algoritme der vælges af flagene under et Strategy interface.

Strategy – 2

- Issue: Forskellen på strategies kan være performance eller kvalitet.
- Issue: Give kontekst med til Strategy, eller lade Strategy objektet hente kontekst efter behov. Det sidste kræver at der er interface til det.
- Issue: Kan have default strategy.

Builder – 1

- Bruges til: At konvertere mellem forskellige repræsentationer af komplekse datastrukturer.
- Eksempler: Persistens. Dokument indlæsning.
- Pointe: At lade en director benytte et generelt Builder interface, der implementeres af forskellige konverteringsklasser. Skiller parsning fra generering.

Til næste gang

- Videreudvikle tegnekonference–systemet (testcases og kode). Indtænke eventuelle nye design patterns.
- Læs artikler:
 - An introduction to Design by Contract, Meyer
 - Error Handling Techniques, Kuslich
 - Handling network timeouts in Java
- Nye krav til tegnekonferencsystemet