

Opgaver fredag den 10. september

Opgave 1

Gruppearbejde Head First Design Patterns, kapitel 1

a) Side 1-7

Den oprindelige løsning med nedarvning og interfaces;

- Diskuter de beskrevne løsninger: den med nedarvning og den anden med interfaces (for at sikre at alle i gruppen har forstået de to løsninger)
- Hvorfor er disse løsninger ikke gode løsninger i eksemplet Duck? (i mange andre situationer er nedarvning og/eller interfaces en god løsning)

b) Side 8-21

Diskuter isolering af det der ændrer sig fra det der er konstant

Den nye løsning med delegation:

- Diskuter den nye løsning
- Hvilke designprincipper anvendes? Formuler disse med egne ord
- Udpeg de vigtigste dele i den færdige kode
- Hvordan kan opførslen for "the ducks" ændres på kørselstidspunktet?

c) Side 22-24

Strategy Pattern:

- Indsæt IS-A, HAS-A og Implements på pilene i diagrammet side 22
- Kommentér Design Principle "Favor composition over inheritance" side 23
Formuler det med egne ord
- Forklar Strategy Pattern side 24

d) Side 25-32

Sammendrag af kapitlet:

- Løs Design Puzzle på side 25
- Formuler med egne ord en opsummering af siderne 26-28

Opgave 2

Gruppearbejde Head First Design Patterns Singleton

a) Side 171-189

Singleton Pattern

- Hvad er det for et problem Singleton Pattern løser?
- Beskriv hvordan Singleton Pattern implementeres i praksis
- Hvad er problemet med Singleton Pattern og tråde?
 - Beskriv hvordan dette kan løses

b) Counter

Skriv en klasse Counter med et Singleton pattern. Klassen skal have en instansvariabel *value*, som initialiseres til 0. Klassen skal desuden have fire metoder:

- `count()` som tæller *value* op med 1
- `times2()` som fordobler *value*
- `zero()` som nulstiller *value*
- `getValue()` som returnerer værdien af *value*

Skriv en anden klasse `AnvendCounter` som anvender Counter.

c) Personadministrator

Lav en `PersonAdministrator` klasse, der har et Set af personer (fra en `Person` klasse du selv finder på) samt metoderne:

- `add(Person person)`
- `getPersoner()`
- `remove(Person person).`

Gør dernæst klassen `PersonAdministrator` til en Singleton, så der kun kan være et objekt af klassen. Skriv en anden klasse `AnvendPersonAdministrator` som anvender `PersonAdministrator`.