Marker som gennemført

Når vi arbejder med de gode programmeringsvaner, koder vi ud fra det koncept, at opnå høj samhørig og lav kobling.

Koblingen mellem klasser handler om deres indbyrdes afhængigheder og samhørigheden handler om hvilke ansvar en klasse har.

Kobling

Hvis vi skal se på et real-life eksempel, så er en IPod (Kan du huske den?) et godt eksempel på en meget tæt kobling.

IPods og andre Appleprodukter har en stor svaghed og det er, at deres batterier hurtigt mister kraften. Når man i sin tid købte sig en IPod, kunne man ligeså godt købe sig en ny når batteriet døde. Batteriet var svejset sammen med printet og kunne ikke udskiftes. Det er et eksempel på en meget tæt kobling, som vi gerne vil undgå når det kommer til programmering.

Samhørighed

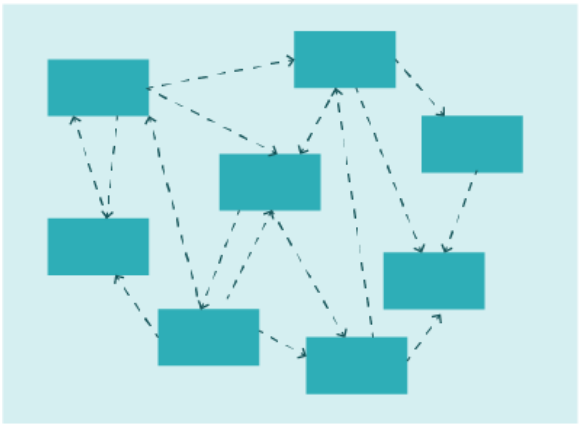
Kan du huske bankreklamen fra Danske Bank?

Danske Bank - Gør det, du er bedst til

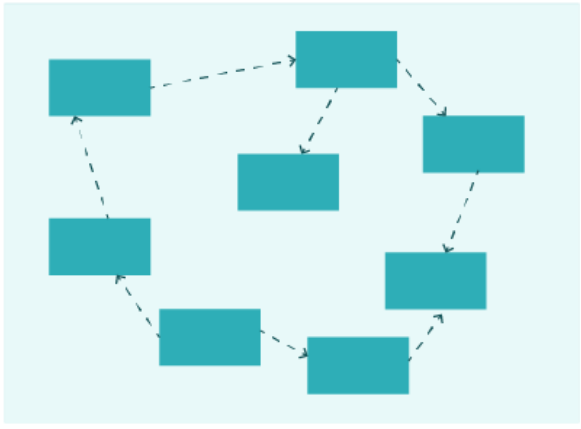
Far from Hollywood

00:45

Deres slogan var: **Gør det du er bedst til - det gør vi**.



Høj kobling



Lav kobling

Når vi snakker samhørighed handler det i bund og grund om at en klasse skal gøre det den er bedst til, så har vi en klasse som skriver til databasen - så skal den ikke også lige pludselig til at sende mail ud og opdatere brugergrænsefladen (Men det ville vi jo aldrig gøre, for vi kender jo 3-lagsmodellen). Når du opretter en ny klasse eller metode, så brug lige 30. sekunder til at tænke på, hvad skal klassen/metoden gøre og en g ? tommelfingerregel er at bruger du et **og eller også** i din dokumentation, så er der noget galt!

```
//Den klasse opretter et brugerobjekt og gemmer objektet i databasen. Den tager sig også af at registrere brugeren til et nyhedsbrev

public class User
{
    ....
}
```

Se evt, videoen

Senest ændret: onsdag den 10. januar 2018, 09:07

◄ Klassediagram

Spring til...

Single responsibility principle (SRP) ►