

Marker som gennemført

Et template pattern bliver tit og alt for ofte overset, fordi man glemmer at benytte sig af alle de fordele man får ved at bruge arv og abstrakte klasser. Et template pattern definerer en række trin i en algoritme og tilbyder subclasses at implementere et eller flere trin alt efter behov.

Lad os kigge nogle brætspil. Det er ligegyldigt om det er skak, ludo eller backgammon (Vi kigger altså op i abstraktionsniveau og finder fællesnævner). Fælles for disse spil er, at der er en række trin der skal følges for at spille.

1. Man skal sætte spillet op og placerer brikker
2. Så kan man starte med at spille
3. Til sidst afsluttes spillet

Lad os kigge lidt på hvordan et template pattern implementeres. Det første der skal gøres er at oprette en abstrakt klasse "BoardGame" og efterfølgende skal de metoder som subclasses skal implementere defineres abstrakt.

```
public abstract class BoardGame
{
    //Denne metode skal implementeres i subclassen
    public abstract void Setup();

    // Denne metode skal implementeres i subclassen
    public abstract void StartPlay();

    // Denne metode udskriver til konsol vinduet at spillet er afsluttet
    public void EndPlay()
    {
        Console.WriteLine("Game ended");
    }
}
```

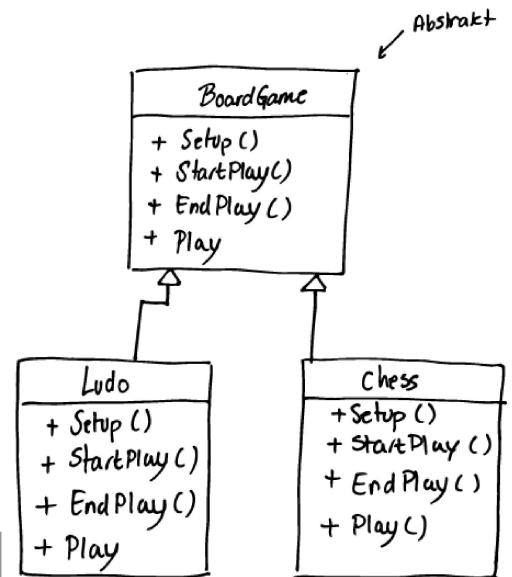
Metoden Play som ligger i BoardGame definerer hvordan algoritmen for spillet ser ud. Først skal brikkerne sættes op og så starter spillet og så afsluttes det. Bemærk at selvom den abstrakte klasse ikke selv implementere de abstrakte metoder, kan den sagtens kalde metoderne. Den abstrakte klasse ved jo at der er en kontrakt mellem den og subclasses, som siger at de SKAL implementere de abstrakte metoder.

```
public void Play()
{
    Setup();
    StartPlay();
    EndPlay();
}
```

Så lad os lige kigge på Ludo klassen, den bliver jo tvunget til at implementere de abstrakte klasser.

```
public class Ludo : BoardGame
{
    public override void Setup()
    {
        Console.WriteLine("Setting up Ludo pieces");
    }

    public override void StartPlay()
    {
        Console.WriteLine("Ludo has started");
    }
}
```



Hvis vi skal starte et ludo spil, kan vi gøre det således

```
BoardGame ludo = new Ludo();  
ludo.Play();
```

Resultat

```
Setting up Ludo pieces  
Ludo has started  
Game ended
```

Skakspillet

```
public class Chess : BoardGame  
{  
    public override void Setup()  
    {  
        Console.WriteLine("Setting up Chess pieces");  
    }  
  
    public override void StartPlay()  
    {  
        Console.WriteLine("Chess has started");  
    }  
}
```

Resultat

```
Setting up Chess pieces  
Chess has started  
Game ended
```

Senest ændret: tirsdag den 9. januar 2018, 15:22

◀ Opgave adapter : Fight Arena

Spring til...

Opgave template : Varme drikke ►