Programmering 2, E2010. Aflevering 3

Mathias Rav, 20103940 Michael Søndergaard, 20104223 DAT-3

Uge 3, November 19, 2010

1 Anonyme klasser og hjælpemetoder

I stedet for anonyme klasser til brug som ActionListeners, har vi valgt at lave en subklasse af JButton, som vi har kaldt ColorButton. Vi kunne dog have brugt en hjælpemetode og anynom klasse til at opnå det samme.

Vi har valgt at gøre det på denne måde, fordi koden bliver pænere, og nemmere at forstå. Hvis vi en dag skulle udvide programmet, ville det være nemt for os, at genbruge koden, som vi allerede har skrevet, andre isteder i programmet.

2 Interfaces

I vores program bruger vi to interfaces. BallIcon og CompositeIcon implementerer begge Icon, og vores ColorButton implementerer ActionListener.

3 Designmønstre

Ud over *Composite*, som vi har implementeret i CompositeIcon, lægger Javas indbyggede UI-biblioteker op til, at vi benytter *Observer* til at reagere på UI-hændelser, sådan som det er implementeret i ColorButton med ActionListener-interfacet.

Strategy-mønstret er brugt til at ændre farven på de enkelte ColorButtons ved at udvide BasicButtonUI-klassen. Java tilbyder mange måder at tegne knapper og andre UI-elementer på gennem sine ComponentUI-klasser. For knapper findes der som standard både BasicButtonUI og MetalButtonUI. Man kan nemt skifte udseendet af sine knapper ved at kalde deres setUI-metode med den passende ButtonUI-instans. I forhold til Strategy-mønstret har vi implementeret en ny strategi til konteksten at tegne knapper - eller rettere, vi har udvidet BasicButtonUI-strategien til at passe til vores formål.

IconCrafter kunne have været opgraderet til en *Facade*, som stod for at håndtere interfacet til JFrame selv, for at gøre arbejdet nemmere for driver-klasserne Driver og IconCrafterProgram. Det har dog ikke vist sig hverken nødvendigt eller bekvemt for et miniprojekt som dette.