Del 1 – TDD

Oppgave 1a – Stack

I denne oppgaven skal vi implementere en enkel Stack. Det skal være mulig å gjøre *Push* for å få elementer på stacken og *Pop* for å få de av. I tillegg ønsker vi funksjonalitet for å se om stacken har noen elementer (f.eks *IsEmpty*) og vi ønsker en funksjon som kan returnere det siste elementet som er lagt på stacken (f.eks *Top*).

Oppgaven skal gjennomføres test-drevet, det betyr at vi følger red/green/refactor prinsippet.

Oppgave 1b – TDD Kata 1

Gjennomfør TDD Kataen som er laget av Roy Osherove (<http://osherove.com/>) som finnes [her.](http://osherove.com/tdd-kata-1/) Kataen går på å implementere en String Calculator, løs èn og èn av oppgavene, prøv å ikke ”se fremover”.

# Oppgave 2 – Logger

I denne oppgaven skal vi implementere en Logger. Det skal være mulig å logge informasjon(*Info*), advarseler (*Warning*) og unntak (*Exception*). Loggeren bør konstruere et format på meldingen som skal skrives, dette formatet bør inneholde dato, klokkeslett, type melding og selvfølgelig meldingen selv. F.eks:

[08.08.2012 15:00][Info] Epost sendt til bruker [arthur@dent.com](mailto:arthur@dent.com).

Denne informasjonen skal skrives til fil.

Oppgave 2b – TDD Kata 2

Gjennomfør TDD Kataen som er laget av Roy Osherove (<http://osherove.com/>) som finnes [her.](http://osherove.com/tdd-kata-2/) Kataen går på å utvide funksjonaliteten String Calculatoren som ble laget i TDD Kata 1, løs èn og èn av oppgavene, prøv å ikke ”se fremover”.

Del 2 TDD i MVC3

Oppgave 3

I denne oppgaven skal vi starte på å lage en ASP.NET MVC 3 Blog applikasjon. Oppgaven skal gjennomføres test-drevet. En bloggpost skal ha en id, tittel og innhold samt dato og klokkeslett for når den ble publisert. Målet med oppgaven er å gjøre det **test-først**.

1. Åpne Visual Studio
2. Opprett et nytt ASP.NET MVC 3 Web application prosject
   1. Bruk **Empty** project template
   2. Bruk **Razor** View Engine
3. Opprett et nytt test prosjekt og legg til en referanse til prosjektet som ble opprettet i 2.
4. Det første vi ønsker er å presentere en oversikt over alle blogposter når siden åpnes. Anta at blogpostene kun lagres i minnet på dette tidspunktet.  
   (*Hint:* *vi trenger en controller med en Index action*).
5. For at blog applikasjonen skal ha verdi må vi kunne lagre blogpostenen. Nå skal vi introdusere et *Repository* som *Controlleren* vi allerede har opprettet kan benytte seg av. Legg merke til at vi fokuserer på at vi skal **hvordan** vi skal hente blogpostene, ikke **hvor**.  
   (*Hint: Vi må gi Controlleren en referanse til et Repository*).