Nogle vink til løsning af den obligatoriske opgave

(Du SKAL ikke følge disse vink, benyt dem kun hvis du ikke kan komme i gang med løsning af opgaven)

- Framework'et skal kunne **Konfigureres** fra en konfiguration-fil (uge 6)

Du bør her indlæse en XML fil og aflæse MaxX hhv maxY for verdenen.
Du kan evt. overføre en parameter i konstruktøren til 'world' klassen, der indeholder stien(path) til konfigurerings filen

- Framework'et skal kunne understøtte **tracing/logging** af beskeder (uge 6)

Til world-klassen kan du lave to metoder der henholdsvis tilmelder og framelder Tracelisteners.
desuden skal du vise en tre-fem steder hvor du benytter logging.

Du kan evt. indkapsle logging i en singleton klasse så der kun er et objekt af logging.
her vil du så kunne gøre det nemmere at logge beskeder ved at have metoder som LogInformation , LogWarning osv.

- Framework'et skal være **dokumenteret** f.eks. ///-kommentarer (uge 5)

Du skal for mindst to klasse lave kommentarer, samt lave doxygen-dokumentation

- Framework'et skal følge principperne fra **SOLID** (uge 11)

Sørg for at hver klasse ikke indeholder for mange forskellige opgaver/data til forskelligt brug (S)

Sørg for at brugeren gennem parametre kan tilpasse og ændre adfærden (behaviour) (O)

Sørg fx for at man ikke kan slå / tilføje negative damage værdier (L)

Sørg for der er interface for mange af klasserne, samt at der ikke er for mange funktioner / metoder i hvert interface (I)

Igen benyt parametre til at kan tilpasse og ændre adfærden gennem Dependency Injection (D)

- Framework'et skal gøre brug af iterationer og **LINQ** (uge 14)

  hvis world har creatures / worldObjects i lister, kan de rette objekter findes gennem find/where eller lign.

  den samlede styrke til angreb hhv. forsvar kan findes genne aggregering (Sum) over våbnene eller finde det mest effektive våben gennem Max eller gennem find alle våben med range over 3

- Framework'et skal indeholde mindst tre **Design Pattern** (uge 5, uge 10, uge 12) f.eks. blandt disse:
  - Template
  - State
  - Composite
  - Observer
  - Decorator
  - Strategy
  - Factory (Abstract Factory)

Som sagt før kan logging være singleton (måske kan konfigurering også være det)

Creature kan være en template, samt kunne understøtte observer, der kunne også være forskellige states

Våben (attack eller defence) kunne være understøttet af decorator og / eller composite

Desuden kunne man lave en factory til at generere tilfældige våben

Verdenen kunne måske også være singleton (der er kun en verden)