## What is the difference between an interface and an abstract class?

## Write down some examples where you would use an interface and some examples where you would use an abstract class

* + En interfaceclass er en kontrakt om at, hvordan en klasse skal se ud. F.eks. er det sådan at game objekt har metoder der hedder move og rotate. Der kan man organisere objekter ved at der sige at de er spil objekter, som alle skal have metoderne move og rotate.
  + I abstrakte klasser er du mere specifik i din kode da en abstrakt klasse ikke kan instantieres og derved kun bruges til klassifikation. Hvor en normal klasse sagtens kan instantieres.

## What are the downsides of inheritance? færdig

* + Man skal opdatere alle de klasser der nedarver fra superklassen, hvis man tilføjer en ny attribut i superklassens konstruktør

## How do I get from a customer describing a system he wants to have to a fully coded system? What steps would you take? Færdig!

1. Noter alt hvad kunden siger han har brug for i et system og stille spørgsmål undervejs. Jeg vil skrive det op på en liste som hedder ”ønsker og krav”
2. Derefter opstille en domænemodel så jeg kunne få en bedre ide om hvilke klasser jeg skal bruge i mit program f.eks. som costumer eller event.
3. Skrive synopse og hvordan jeg gerne ville have at programmet skulle virke.
4. Opstille et klassediagram med attributter og metoder.
5. Oprettelse af et sekvensdiagram, så jeg har en ide om hvordan programmets klasser skal interagere med hinnanen og med brugeren.
6. Begynde at skrive koden og derefter gennemgå den igen, tjekke den for fejl, optimere koden så den kan komme til at køre mere effektivt og gøre den mere læsbar af udviklere.

## What is the difference between coding and developing software? færdig

* + Når man koder software, så skriver man bare koden uden at tænke over den større helhed i programmet
  + Når man udvikler software, så bruger man diagrammer, tænker over hvordan man brugeren bruger programmet og prøver at gøre koden mere læsbar.

## What do you think is good code? Færdig!

* Kode som er nem at læse
* Kode som ikke bruger unødvendige metoder/attributter
* Kode med gode variabel navne
* Der må ikke være kommentarer der ikke er relevante for koden
* Metoder skal generelt være så simple som muligt, siden de meget nemt kan blive meget lange
* Der må ikke være noget ud kommenteret kode
* Ingen magiske tal, tal uden begrundelse