

Programmering og Problemløsning

13.1: Objektorienteret design

Klasser og objekter:

Header, constructor, fields, functions, and members

```
type vehicle (w : int) =  
  do printfn("Creating a vehicle object");  
  let _wheels = w;  
  member this.wheels = _wheels  
  
let car = vehicle 4;  
let bike = vehicle 2;  
printfn "A car has %d wheels, a bike has %d wheels" car.wheels bike.wheels
```

```
sporring@Jons-mac 13.1-3 % fsharp vehicle.fsx  
Creating a vehicle object  
Creating a vehicle object  
A car has 4 wheels, a bike has 2 wheels
```

vehicle

Klasser og objekter: Accessors

```
type vehicle (w : int) =  
  do printfn("Creating a vehicle object");  
  let mutable _wheels = w;  
  member this.wheels  
    with get () = _wheels  
    and set (n) = _wheels <- n  
  
let car = vehicle 4;  
printfn "A car has %d wheels" car.wheels  
car.wheels <- 6;  
printfn "A modded car has %d wheels" car.wheels
```

```
sporrings@Jons-mac 13.1-3 % fsharp vehicleAccessors.fsx  
Creating a vehicle object  
A car has 4 wheels  
A modded car has 6 wheels
```



vehicle

Klasser og objekter: Nedarvning

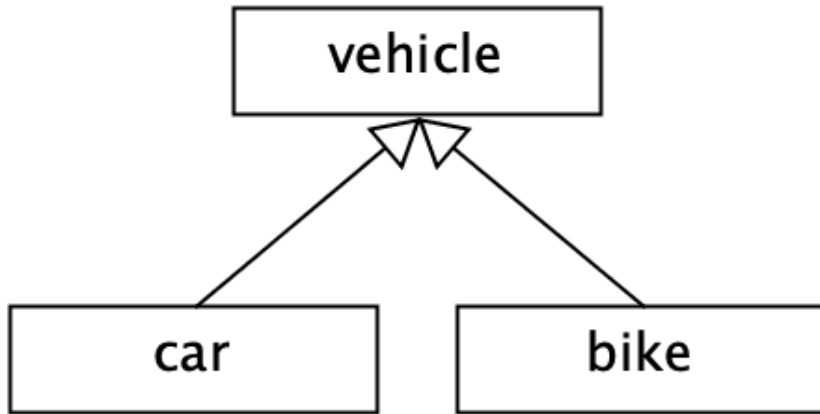
```
type car (f : float) =  
  do printfn("Creating a car object");  
  member this.wheels = 4  
  member this.fuelEfficiency = f  
  
type bike (w : float) =  
  do printfn("Creating a bike object");  
  member this.wheels = 2  
  member this.weight = w  
  
let car = car 21.5;  
let bike = bike 15.2;  
printfn "A car has %d wheels and runs %.2f km/l" car.wheels car.fuelEfficiency  
printfn "A bike has %d wheels and weighs %.2f kg" bike.wheels bike.weight
```

```
[sporrings@Jons-mac 13.1-3 % fsharp carNBike.fsx  
Creating a car object  
Creating a bike object  
A car has 4 wheels and runs 21.50 km/l  
A bike has 2 wheels and weighs 15.20 kg
```

car

bike

Klasser og objekter: Nedarvning



```
type vehicle (w : int) =  
  do printfn("Creating a vehicle object");  
  member this.wheels = w
```

```
type car (f : float) =  
  inherit vehicle(4)  
  do printfn("Creating a car object");  
  member this.fuelEfficiency = f
```

```
type bike (w : float) =  
  inherit vehicle(2)  
  do printfn("Creating a bike object");  
  member this.weight = w
```

```
let aVehicle = vehicle 3  
let car = car 21.5  
let bike = bike 15.2  
printfn "A vehicle has %d wheels" aVehicle.wheels  
printfn "A car has %d wheels and runs %.2f km/l" car.wheels car.fuelEfficiency  
printfn "A bike has %d wheels and weighs %.2f kg" bike.wheels bike.weight
```

```
sporrings@Jons-mac 13.1-3 % fsharp vehicleInherit.fsx  
Creating a vehicle object  
Creating a vehicle object  
Creating a car object  
Creating a vehicle object  
Creating a bike object  
A vehicle has 3 wheels  
A car has 4 wheels and runs 21.50 km/l  
A bike has 2 wheels and weighs 15.20 kg
```

Design efter navne- og udsagnsord

Navneord er kandidater til klasser, udsagnsord er kandidater til metoder, som får klasserne til at hænge sammen.

Navneord (substantiv): Genstande, levende væsner, personer eller abstraktioner.
F.eks.: (et) bord, (en) mus, Jon, (et) venskab.

Udsagnsord (verbum): Handlinger, hændelser, tilstandsmåder.
F.eks.: (at) løbe, (han) underviser, (han) er (lærer)

Design efter navne- og udsagnsord

Use-case: Sænke slagskibe

Dette er et spil for to personer, der kan spilles med papir og blyant. Der spilles på fire plader, to for hver spiller, og hver plade er inddelt i 10x10 felter. Hvert felt identificeres vha. dets række og søjle nummer.

Hver spiller får tildelt et antal skibe, som placeres på spillerens ene plade og markerer, hvor modstanderen har forsøgt at skyde. På den anden plade markerer spilleren tilsvarende, hvor han/hun har forsøgt at ramme modstanderen.

Når skibene er placeret skiftes spillerne til at skyde på modstanderens felt, og modstanderen annoncerer ramt eller plask, alt efter om et skib blev ramt eller ej. Vinderen er den, der først får sænket alle modstanderes skibe.

Design efter **navne**- og udsagnsord

Use-case: Sænke slagskibe

Dette er et **spil** for to **personer**, der kan spilles med **papir** og **blyant**. Der spilles på fire **plader**, to for hver **spiller**, og hver **plade** er inddelt i 10x10 **felter**. Hvert **felt** identificeres vha. dets **række-** og **søjlenummer**.

Hver **spiller** får tildelt et antal **skibe**, som placeres på **spillerens** ene **plade** og markerer, hvor **modstanderen** har forsøgt at skyde. På den anden **plade** markerer **spilleren** tilsvarende, hvor han/hun har forsøgt at ramme **modstanderen**.

Når **skibene** er placeret skiftes **spillerne** til at skyde på **modstanderens** **felt**, og **modstanderen** annoncerer ramt eller plask, alt efter om et **skib** blev ramt eller ej. **Vinderen** er den, der først får sænket alle **modstanderes** **skibe**.

Design efter navne- og udsagnsord

Use-case: Sænke slagskibe

Dette er et spil for to personer, der kan spilles med papir og blyant. Der spilles på fire plader, to for hver spiller, og hver plade er inddelt i 10x10 felter. Hvert felt identificeres vha. dets række- og søjlenummer.

Hver spiller får tildelt et antal skibe, som placeres på spillerens ene plade og markerer, hvor modstanderen har forsøgt at skyde. På den anden plade markerer spilleren tilsvarende, hvor han/hun har forsøgt at ramme modstanderen.

Når skibene er placeret skiftes spillerne til at skyde på modstanderens felt, og modstanderen annoncerer ramt eller plask, alt efter om et skib blev ramt eller ej. Vinderen er den, der først får sænket alle modstanderes skibe.

Sænke slagskibe

Navneord: Spil, person, papir, blyant, plade, felt, blyant, papir, række nummer, søjlenummer, spiller, skibe, modstander, (en) vinder.

Udsagnsord: er, spille, inddele, identificere, tildele, placere, forsøge, skyde, markere, ramme, skifte, annoncere, sænke.

Et **spil** består af 2 **spillere**.

Hver **spiller** har 2 **plader**.

Hver **plade** er inddelt i 10x10 **felter**.

Hvert **felt** har et **række-** og et **søjlenummer**.

Hver **spiller** tildeles **skibe**.

Dette er et **spil** for to **personer**, der kan spilles med **papir** og **blyant**. Der spilles på fire **plader**, to for hver **spiller**, og hver **plade** er inddelt i 10x10 **felter**. Hvert **felt** identificeres vha. dets **række** og **søjle** nummer.

Hver **spiller** får tildelt et antal **skibe**, som **placeres** på **spillerens** ene **plade** og **markerer**, hvor **modstanderen** har **forsøgt** at skyde. På den anden **plade** **markerer** **spilleren** tilsvarende, hvor han/hun har **forsøgt** at ramme **modstanderen**.

Når **skibene** er placeret **skiftes** **spillerne** til at skyde på **modstanderens** **felt**, og **modstanderen** **annoncerer** ramt eller plask, alt efter om et **skib** blev ramt eller ej. **Vinderen** er den, der først får sænket alle **modstanderes** **skibe**.

Skibe **placeres** på en **plade**.

Spiller **skyder** på **modstanderen**.

Modstander **annoncerer** ramt eller plask.

Spiller **markerer** **skud**.

Skib kan blive **ramt** og **sunket**.

Vinder er den, som har **sunket** alle **modstanders** **skibe**.

Sænke slagskibe

Navneord: Spil, ~~person~~, ~~papir~~, ~~blyant~~, plade, felt, ~~blyant~~, ~~papir~~, række- og søjle- nummer, spiller, skibe, modstander, (en) vinder.

Udsagnsord: er, ~~spille~~, inddele, identificere, tildele, placere, ~~forsøge~~, skyde, markere, ramme, ~~skifte~~, annoncere, sænke.

Et **spil** består af 2 **spillere**.

Hver **spiller** har 2 **plader**.

Hver **plade** er inddelt i 10x10 **felter**.

Hvert **felt** har et **række-** og et **søjle-**nummer.

Hver **spiller** tildeles **skibe**.

Valg: At skyde eller et skud? At vinde eller en vinder?

Skibe **placeres** på en **plade**.

Spiller **skyder** på **modstanderen**.

Modstander **annoncerer** ramt eller plask.

Spiller **markerer** **skud**.

Skib kan blive **ramt** og **sunket**.

Vinder er den, som har **sunket** alle **modstanders** **skibe**.

Sænke slagskibe

Et spil består af 2 spillere.

Hver spiller har 2 plader.

Hver plade er inddelt i 10x10 felder.

Hvert felt har et række- og et søjlenummer.

Hver spiller tildeles skibe.

Skibe placeres på en plade.

Spiller skyder på modstanderen.

Modstander annoncerer ramt eller plask.

Spiller markerer skud.

Skib kan blive ramt og sunket.

Vinder er den, som har sunket alle modstanders skibe.

Et spil består af spillere.

En spiller har en modstander.

En spiller har skibe.

En spiller har plader.

En plade består af felder.

Et felt har et koordinat (søjle- og række-nummer).

En spiller kan vinde.

En spiller skyder på en modstander.

En spiller markerer skud på en plade.

Et skib kan blive ramt.

Sænke slagskibe: relationer

```
/// A game is a battleship game
type game() = class end

/// A player is a human player, which has 2 boards and
/// several ships
type player() = class end

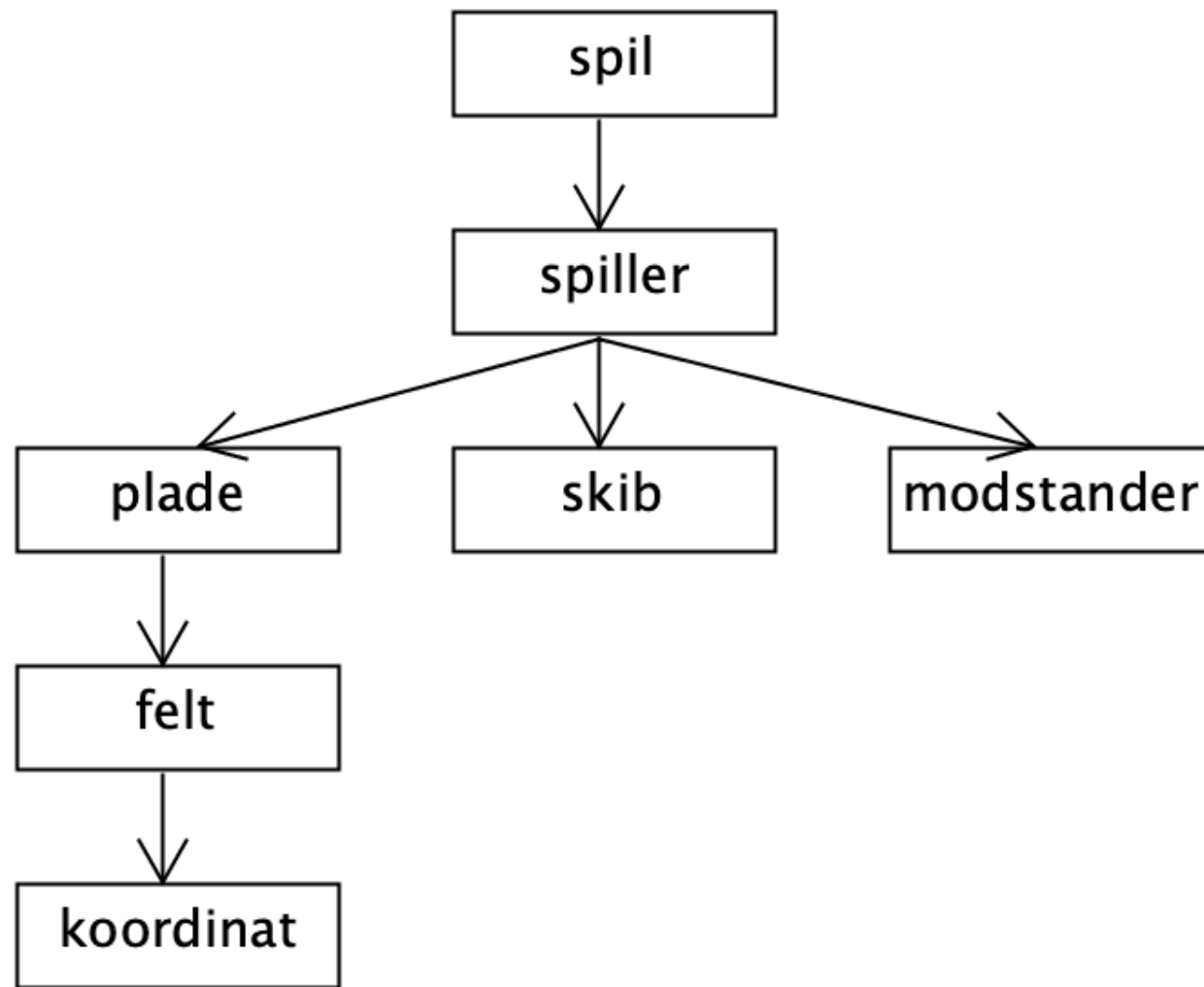
/// An opponent is another player
type opponent() = class end

/// A ship can be damaged
type ship() = class end

/// A board is a square set of fields with row-column
/// coordinates. Ships are placed on the board
type board() = class end

/// A field is can be covered by a ship and can have been
/// shootend at
type field() = class end

/// A coordinate is a location on a board
type koordinat() = class end
```



Sænke slagskibe

```
/// A game is a battleship game
type game() = class end

/// A player is a human player, which has 2 boards and
/// several ships
type player() = class end

/// An opponent is another player
type opponent() = class end

/// A ship can be damaged
type ship() = class end

/// A board is a square set of fields with row-column
/// coordinates. Ships are placed on the board
type board() = class end

/// A field is can be covered by a ship and can have been
/// shootend at
type field() = class end

/// A coordinate is a location on a board
type coordinate() = class end
```

```
/// A coordinate is a location on a board
type coordinate(row : int, col : int) =
  member this.row with get () = row
  member this.col with get () = col

/// A field is can be covered by a ship and can have been
/// shootend at
type field(c : coordinate) =
  member this.coord with get () = c

/// A board is a square set of fields with row-column
/// coordinates. Ships are placed on the board
type board() =
  let fields = Array2D.init 10 10 (fun i j -> field (coordinate (i, j)))

/// A ship can be damaged
type ship() = class end

/// An opponent is another player
type opponent() = class end

/// A player is a human player, which has 2 boards and
/// several ships
type player() =
  let my_board = board()
  let opponents_board = board()

/// A game is a battleship game
type game() =
  let p1 = player()
  let p2 = player()
```