

Design af genbrugeligt objektorienteret software

– Evaluering af software ved hjælp af statistiske mål.

24 februar 2004

Specialeforsvar af:

Søren Gaardbo Jensen

- Hvad er problemet?

- Hvad er problemet?

”Hvad karakteriserer ’det gode design’ i forhold til genbrug og vedligeholdelse, og kan vi automatisk evaluere software så problematiske dele identificeres?”

- Hvad er problemet?

”Hvad karakteriserer ’det gode design’ i forhold til genbrug og vedligeholdelse, og kan vi automatisk evaluere software så problematiske dele identificeres?”



- ”Reusability/Maintainability Profiler”.
- Genbrug? - Vedligeholdelse?

- Strategi.
 - Eksisterende softwaremål.
 - Designprincipper og sammenhæng med softwaremålene.
 - Tilstrækkelighed i mål.
 - Eksperimenter.

- Objektorienterede paradigme – Java.

- Objektorienterede paradigme – Java.
- Ingen metadata.

- Objektorienterede paradigme – Java.
- Ingen metadata.
- Statiske mål.

```
public class Receipt {  
    private Sale sale;  
  
    public Receipt(Sale _sale) {  
        sale = _sale;  
    }  
  
    public String getText() {  
        float total=0;  
        String receipt;  
  
        receipt = sale.getCustomer().getName() + "\n";  
        receipt += sale.getCustomer().getAddress() + "\n\n";  
  
        SalesItems salesItems = sale.getSalesItems();  
        Product[] p = salesItems.getProducts();  
  
        for(int i=0; i<salesItems.getProductCount();i++) {  
            receipt += p[i].getName() + "\t" + p[i].getPriceInclVat() + "\n";  
            total += p[i].getPriceInclVat();  
        }  
        receipt += "\ntotal:\t\t" + total + "\n";  
        return receipt;  
    }  
}
```

Software mål

CBO = 5
LOC = 18

- Plug-in til Eclipse blev udviklet.
- Sammenhæng mellem nogle mål og principper.
- Nye mål blev konstrueret:
 - Abstraktionsfaktorer.
 - Kendskab.

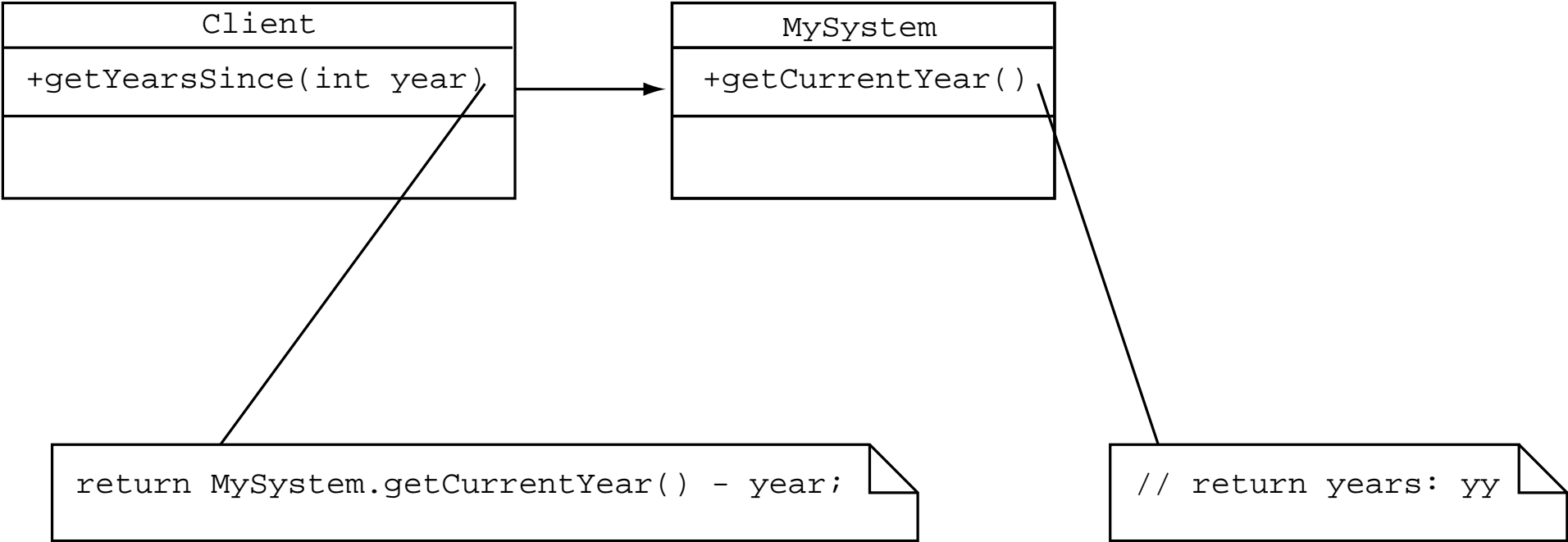
- Målene kan i nogen grad identificere potentielt problematisk kode.
 - Kan ikke bruges helt uden metadata.
- Abstraktionsfaktorene ser lovende ud.
 - Skal undersøges nærmere.

- Eksisterende softwaremål.
- Designprincipper og sammenhæng med softwaremålene.
- Tilstrækkelighed i mål.
- Eksperimenter.
- Konklusion.

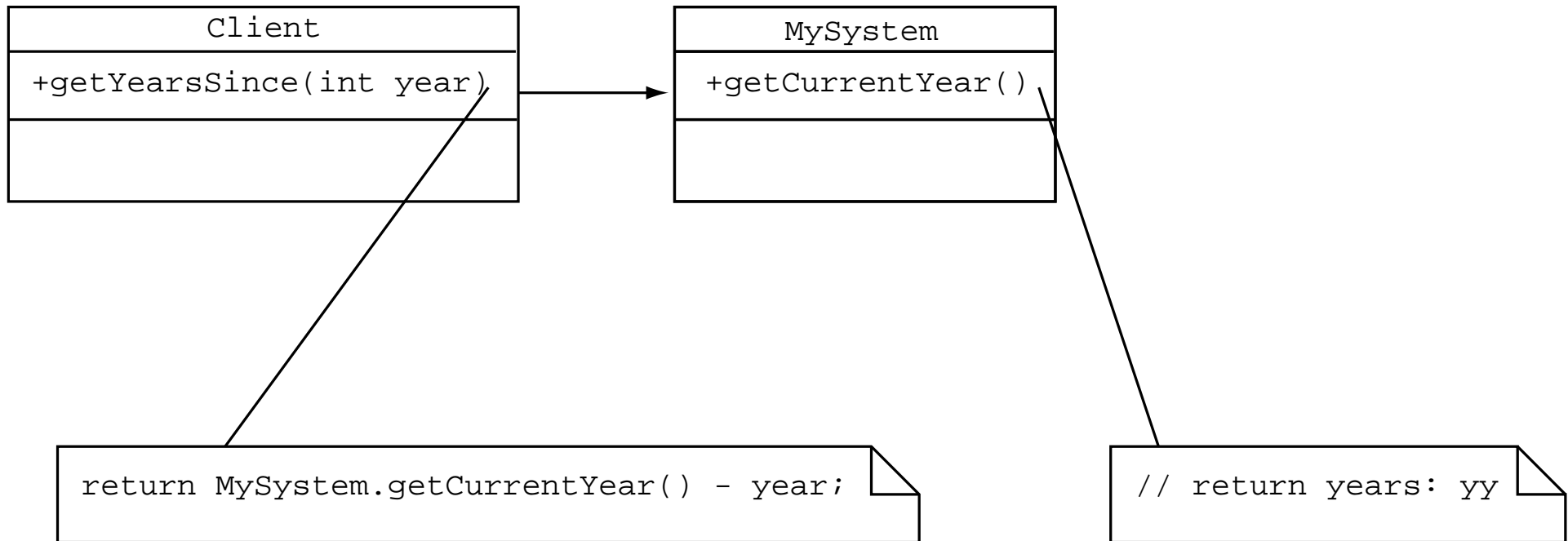
- Strukturerede paradigme.
- Objektorienterede paradigme.
- To gennemgående begreber:
 - Kobling.
 - Kohæsion.

- Kobling.

- Kobling.



- Kobling.

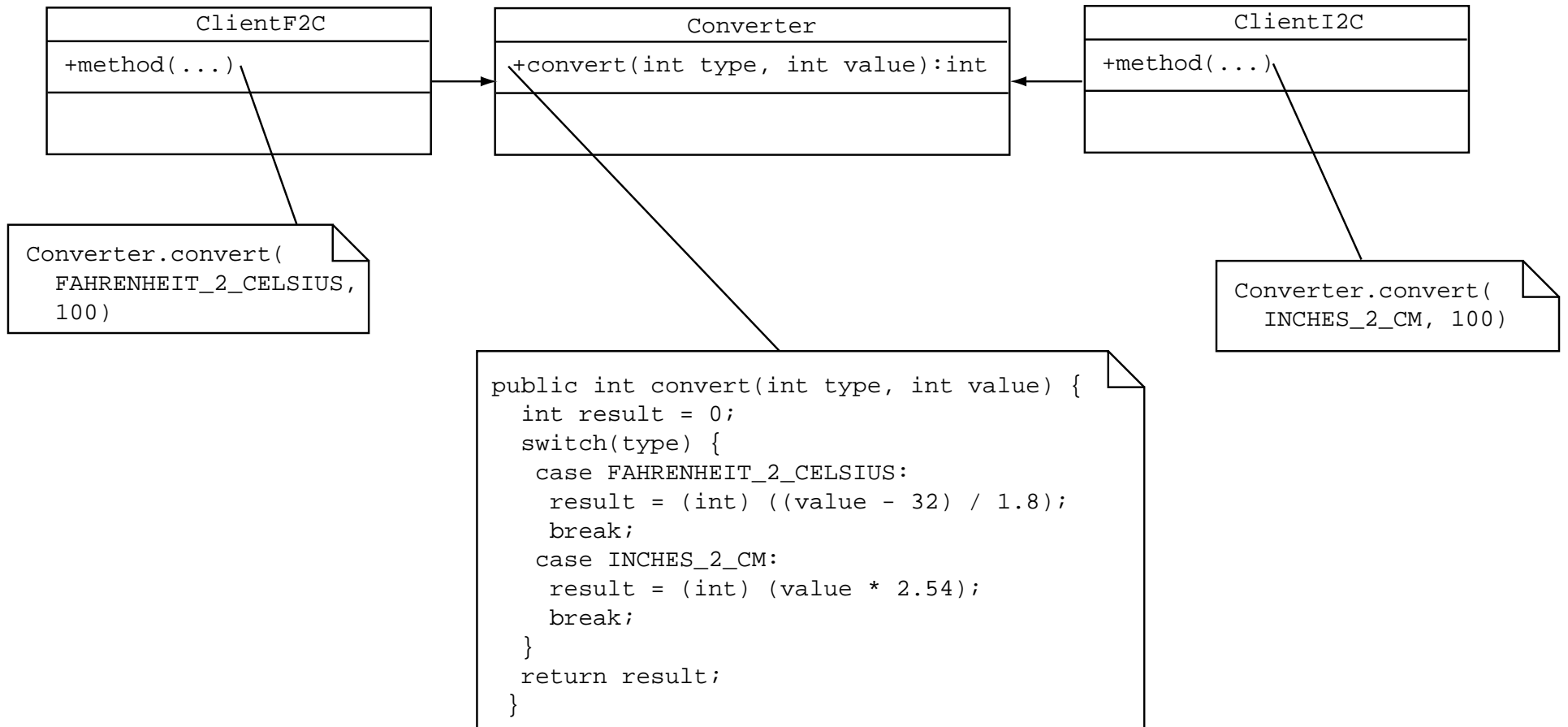


- Kobling fra client til MySystem.

- Ændringer i et modul kan kræve ændringer i koblede moduler.
- Svært at isolere til genbrugelige moduler.
- Svært at overskue: Kræver kendskab til koblede dele.

- Kohæsion.

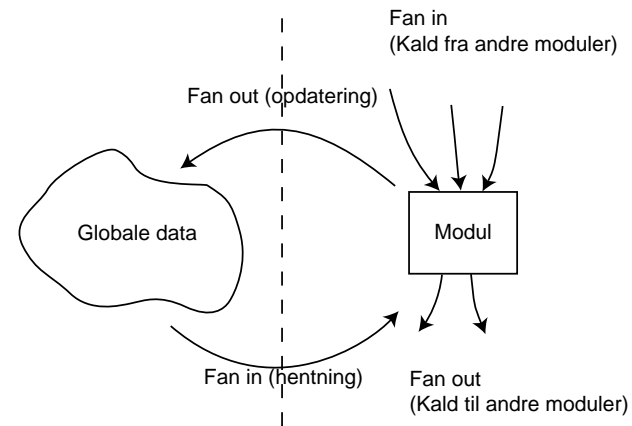
• Kohæsion.



- Lettere at vedligeholde.
- Mindre fejlbehæftet [McConnell]
- Lav kohæsion er udtryk for uhensigtsmæssigt design.

Måling af kobling

- Fan in/fan out.
- Coupling Between Object classes: *CBO*
- Coupling Factor: *COF*.



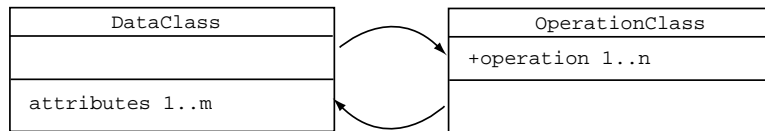
- Kan ikke måles!

- Kan ikke måles!
- Lack of COhesion in Methods: *LCOM*.
 - Flere versioner.

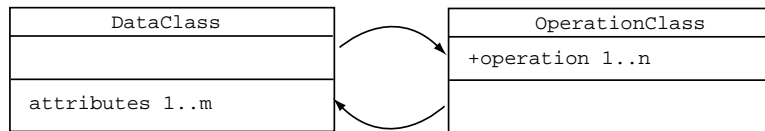
- Eksisterende softwaremål.
- **Designprincipper og sammenhæng med softwaremålene.**
- Tilstrækkelighed i mål.
- Eksperimenter.
- Konklusion.

- Der findes mange principper.
- Mange egenskaber kan ikke måles.
- Udvalgt fire principper:
 - Information Expert.
 - Polymorphism.
 - Law of Demeter.
 - Design mod et interface.

- Information Expert.
 - Mindre kommunikation mellem klasser.
 - Højere kohæsion i klassen.

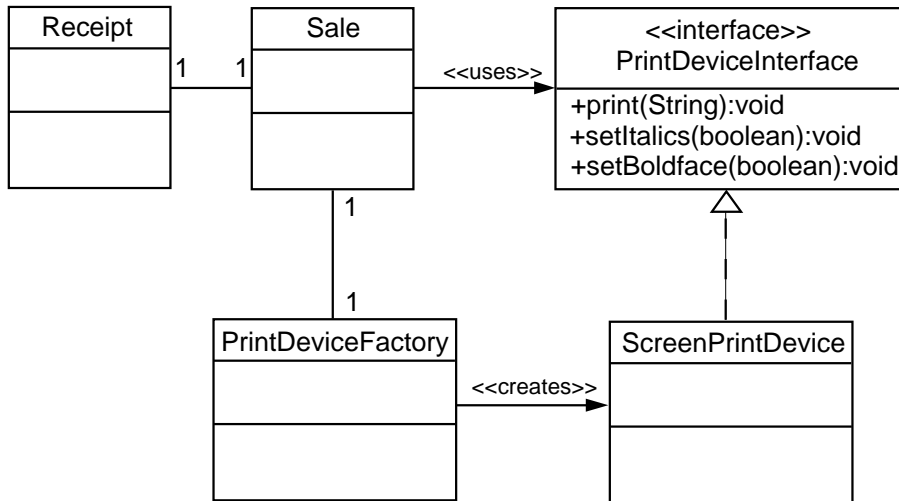


- Information Expert.
 - Mindre kommunikation mellem klasser.
 - Højere kohæsion i klassen.
- Ikke problemfrit.



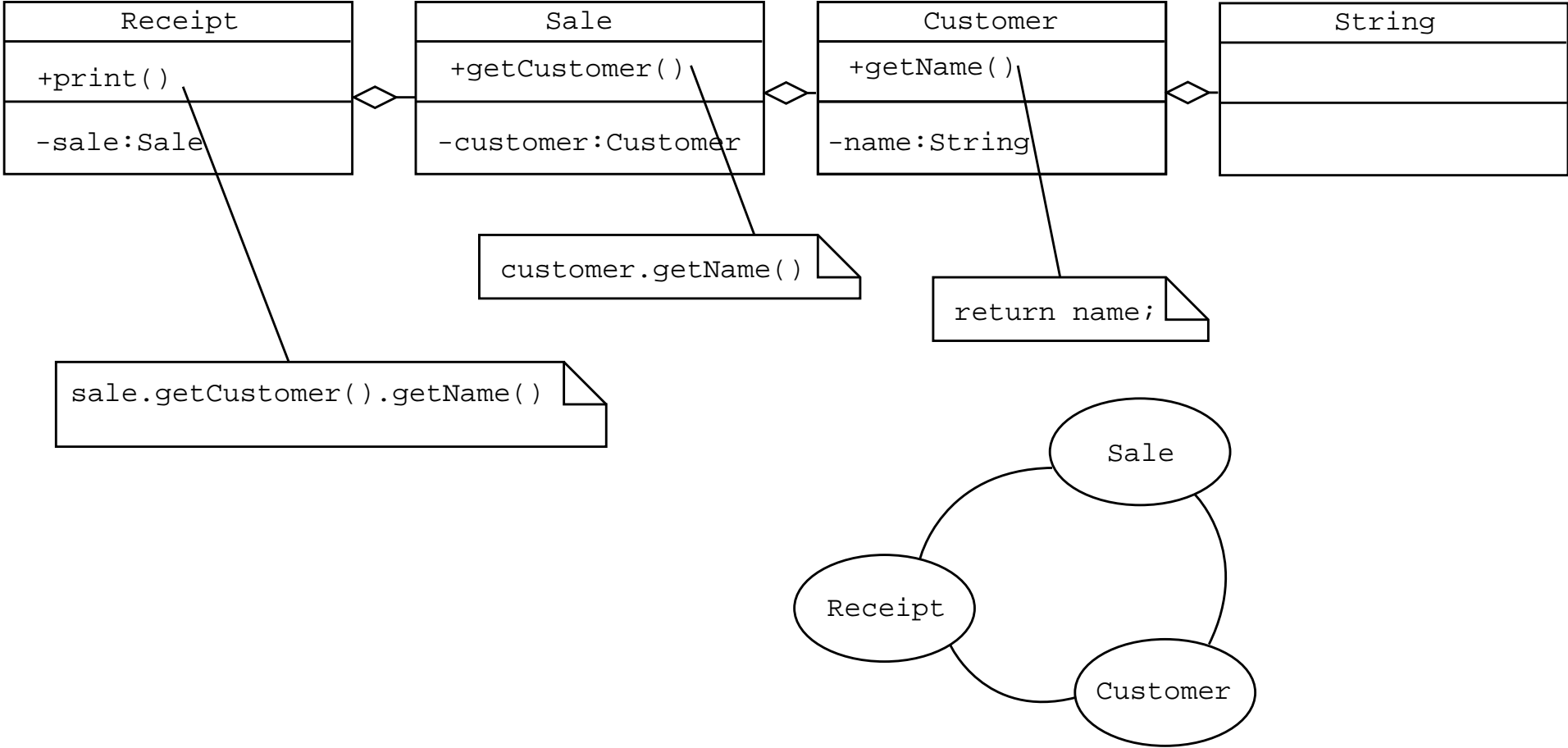
Polymorphism

- Polymorphism.
 - Bruges til udvidelser og erstatninger.
 - Hvis princippet erstatter 'switch' i kode, reduceres koblingen.

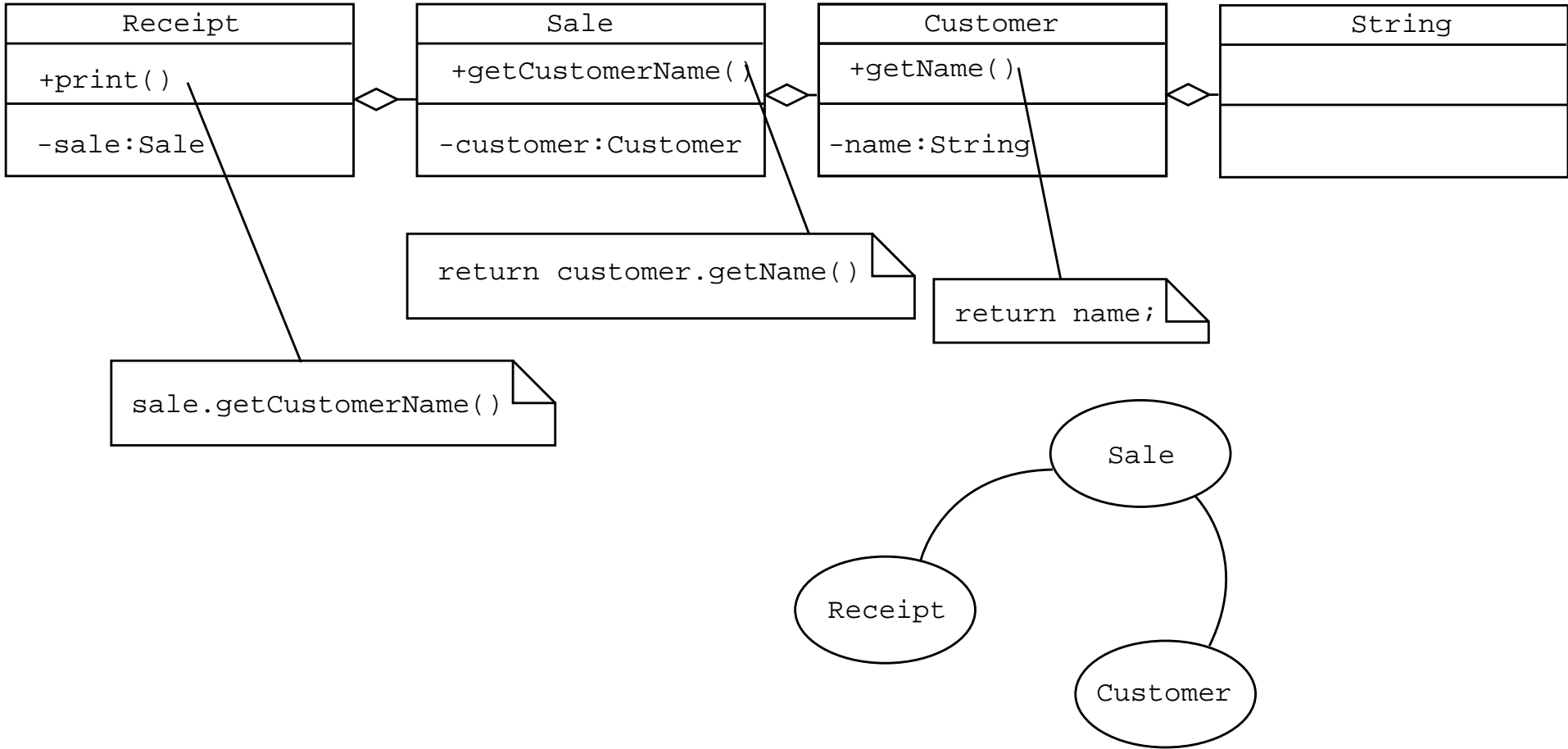


Law of Demeter

- Law of Demeter.



Law of Demeter - 2



• Reducerer koblinger i systemet.

- Design mod et interface.
 - Generelt princip.
 - Sikrer *information hiding*.
 - Polymorphism afhænger af dette.

- Eksisterende softwaremål.
- Designprincipper og sammenhæng med softwaremålene.
- **Tilstrækkelighed i mål.**
- Eksperimenter.
- Konklusion.

- Ingen mål afslører brud på "Design mod et interface".

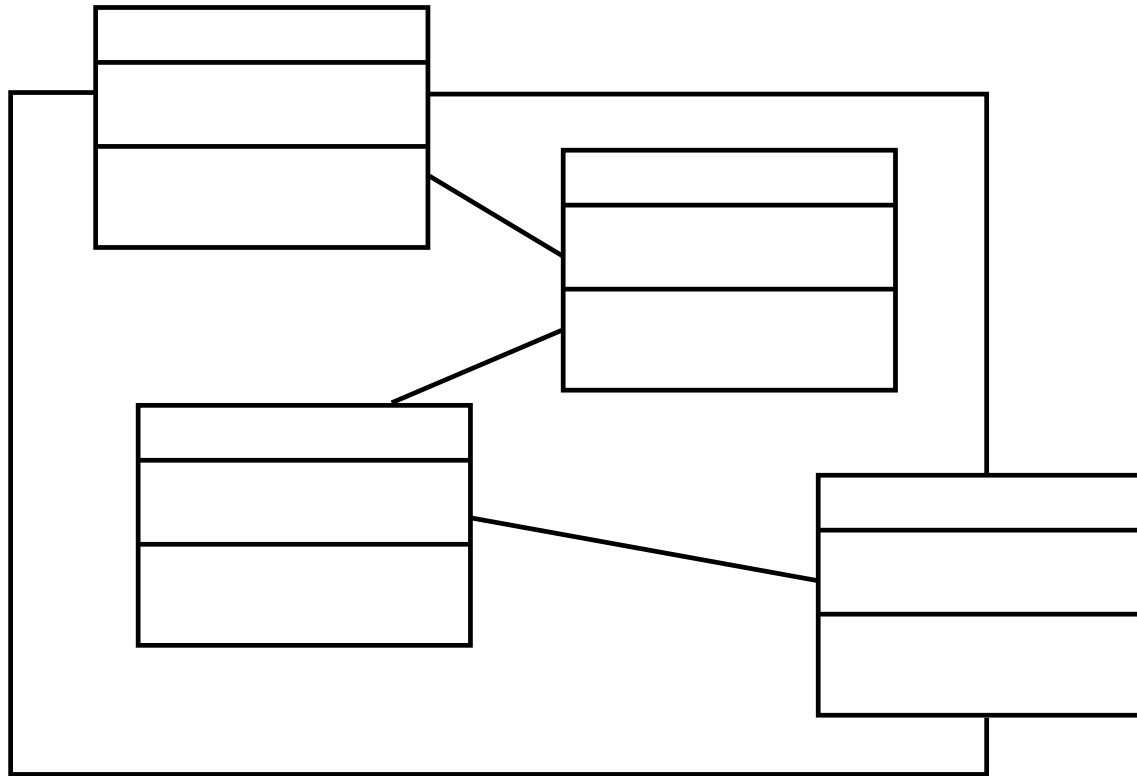
- Ingen mål afslører brud på "Design mod et interface".
- Abstraktionsfaktorer.

- Ingen mål afslører brud på "Design mod et interface".
- Abstraktionsfaktorer.
 - KoblingsAbstraktionsFaktor baseret på *CBO*.

- Koblingsbaseret abstraktion er ikke altid tilstrækkelig.

Abstraktionsfaktorerne

- Koblingsbaseret abstraktion er ikke altid tilstrækkelig.



- Kendskab.
 - Tæller samtlige typer der er kendskab til.
 - Supermængde til *CBO*.
 - "Grænseklasser" medtages i målet.

- Forholdet mellem `interface` koblinger i forhold til samtlige koblinger.
- Forholdet mellem `interface` kendskaber i forhold til samtlige kendskaber.

- Eksisterende softwaremål.
- Designprincipper og sammenhæng med softwaremålene.
- Tilstrækkelighed i mål.
- **Eksperimenter.**
- Konklusion.

- To forskellige eksperimenter.
- Tre softwaremål
 - *LCOM**.
 - *CBO*.
 - *Kendskabet*.
- Design mod et interface.
 - Abstraktionsfaktorene.

- Information Expert.
 - Kobling er reduceret.
 - Kohæsion er forbedret (småt system).
 - Kendskab er reduceret.

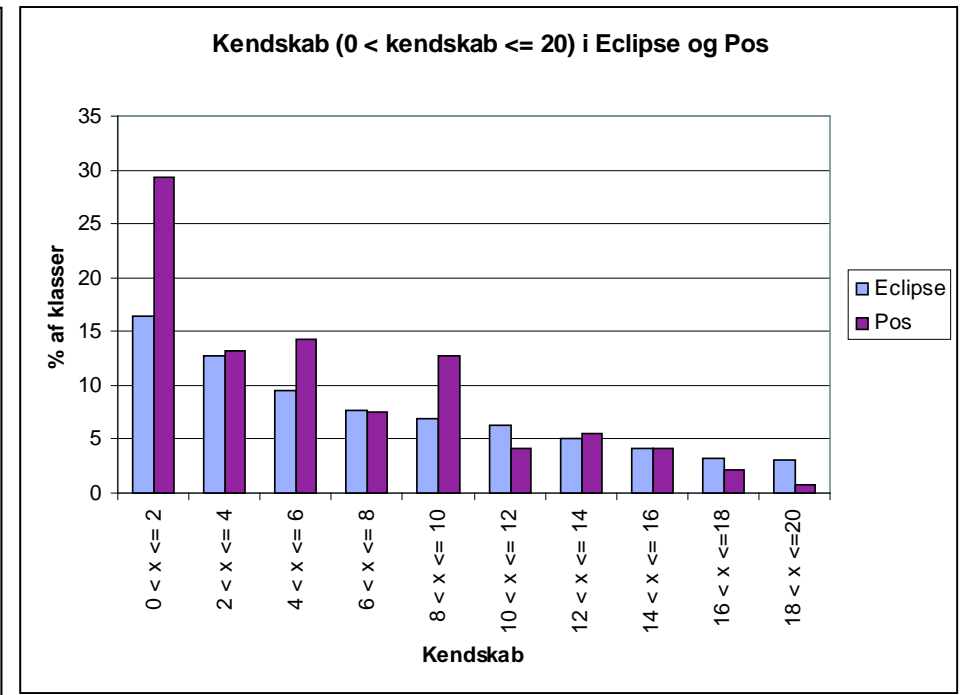
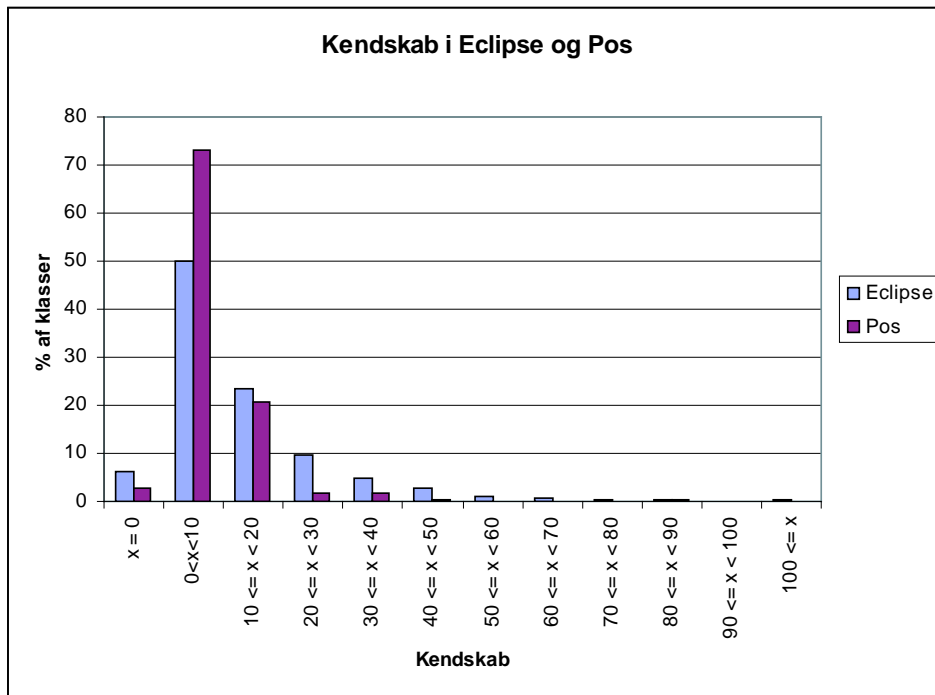
- Polymorphism.
 - Koblingen er "bedre" pga. omfordeling af koblingerne.
 - Kohæsionen er uændret (småt system).
 - Kendskab er reduceret.

- Law of Demeter.
 - Kobling er "bedre" pga. omfordeling af koblingerne.
 - Kohæsionen er uændret (småt system).
 - Kendskab er lidt reduceret uden parm. og lidt forøget med parm.

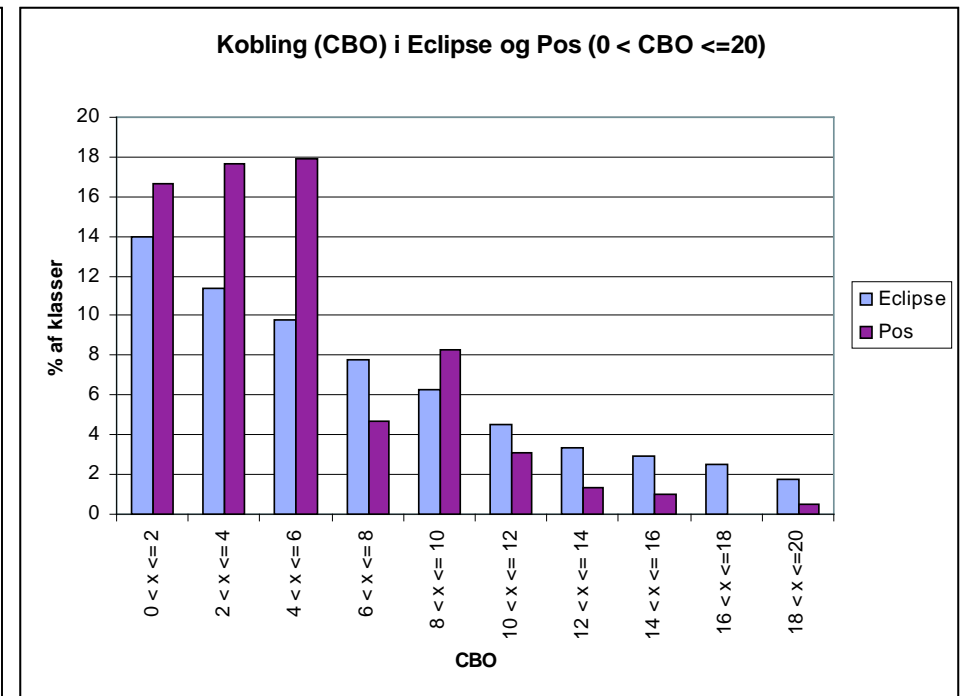
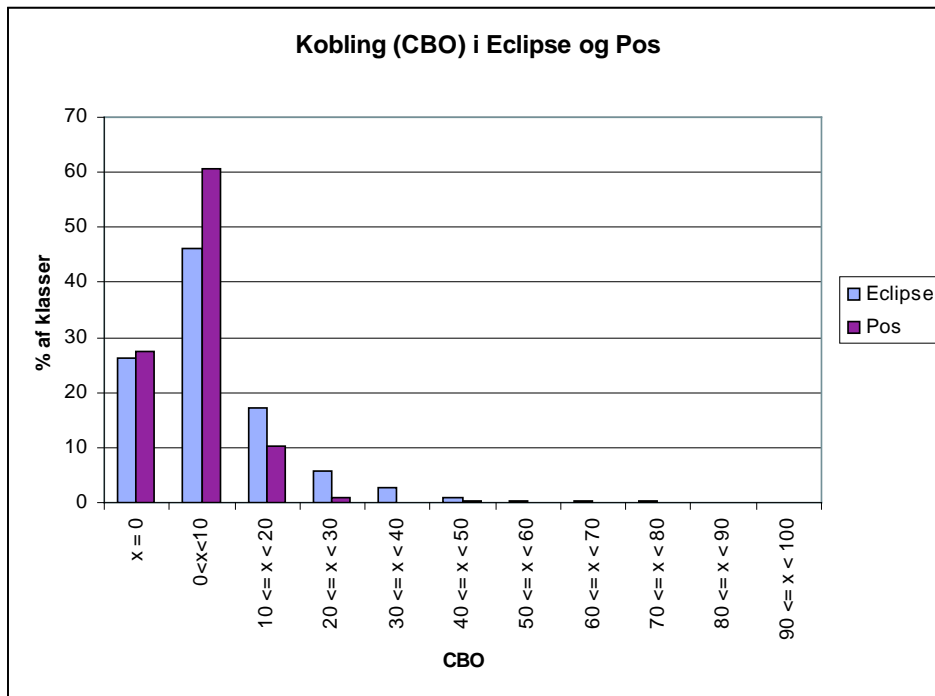
- Svært at sige noget om kohæsionen pga. systemets størrelse.
- Generelt bliver koblinger reduceret.
- Generelt bliver kendskabet reduceret.
- Koblingsmålet kan måske bruges til identifikation af problematiske klasser.

- Interfaceprincippet i Eclipse & POS.
- To versioner: Kendskab & kobling.
- Forventet at Eclipse er mere "abstrakt" end POS.

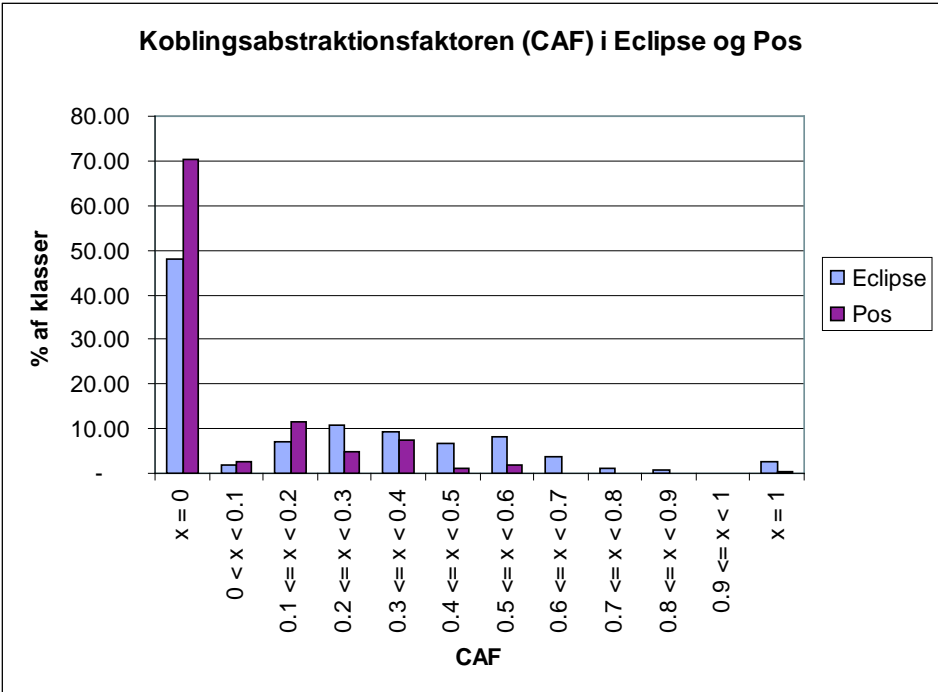
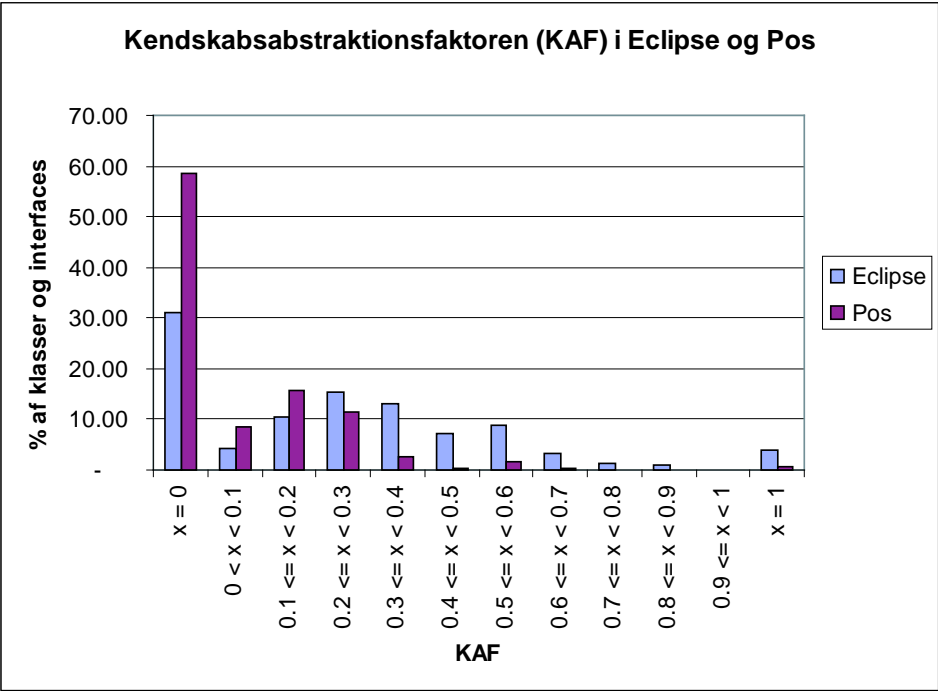
● Kendskab.



● Kobling.



● Abstraktionsfaktorerne.



- Eksisterende softwaremål.
- Designprincipper og sammenhæng med softwaremålene.
- Tilstrækkelighed i mål.
- Eksperimenter.
- **Konklusion.**

- Der er en sammenhæng mellem designprincipper og mål.
- En automatisk identifikation af potentielt problematiske områder er mulig men...

- Der er en sammenhæng mellem designprincipper og mål.
- En automatisk identifikation af potentielt problematiske områder er mulig men...
 - ...der er brug for "metadata" for at kunne anvende resultaterne.

- Der er en sammenhæng mellem designprincipper og mål.
- En automatisk identifikation af potentielt problematiske områder er mulig men...
 - ...der er brug for "metadata" for at kunne anvende resultaterne.
- Eksperimenterne var for små til at sige noget om målet *LCOM**.

- Der er en sammenhæng mellem designprincipper og mål.
- En automatisk identifikation af potentielt problematiske områder er mulig men...
 - ...der er brug for "metadata" for at kunne anvende resultaterne.
- Eksperimenterne var for små til at sige noget om målet *LCOM**.
- Abstraktionsmålene forventes at sige noget om systemets generelle fleksibilitet, men skal undersøges nærmere.