

2. Indkapsling

- **Definition:** At skrive kode, der skjuler unødvendige detaljer for andre moduler og ikke afhænger af andre modulers implementeringer.
- **Fordele:** Forbedret modularitet, lettere vedligeholdelse, bedre fejlfinding og sikkerhed.
- **Ulemper:** Kan føre til overkompleksitet og udfordringer med debugging.
- **Princip:** "Detaljer, der kræves for at returnere et resultat fra en klassemetode, bør skjules fra den kaldende metode."

3. Eksempel på indkapsling: Evil Client

- **Problem:** En klient kan ødelægge data ved direkte adgang til en classes interne struktur.
- **Løsning:** Brug af private felter og offentlige metoder til at manipulere data sikkert.

4. Abstraktion

- **Definition:** At skabe et nyt semantisk niveau, hvor man kan være præcis.
- **Formål:** Reducere kompleksitet ved at skjule irrelevante detaljer og fremhæve de relevante.
- **Eksempel:** Forskellige repræsentationer af en person afhængigt af konteksten (f.eks. studerende, ven, kontor).

5. Arv

- **Definition:** En "is-a" relation mellem klasser, hvor en underklasse arver egenskaber fra en overklasse.
- **Koncepter:**
 - **Specialisering:** Underklassen har specifikke egenskaber.
 - **Generalisation:** Overklassen har generelle egenskaber.
- **Princip:** Liskov's Substitution Principle – en underklasse kan erstatte en overklasse uden at ændre programmets korrekthed.
- **Quiz:** Klassificer fugle (f.eks. Pingvin og Svale) i en arvshierarki og vurder hierarkiets gyldighed ved at tilføje metoder (f.eks. LayEgg(), Fly()).

6. Polymorfi

- **Definition:** Evnen for objekter at tage mange former afhængigt af deres type.
- **Brug:** Nødvendig, når objekter skal have type-specifik adfærd.
- **Eksempel:** I et real-time strategi spil kan en enhed reagere forskelligt på en fjende afhængig af sin type (f.eks. angribe, forsvare, flygte).

Generelle tips:

- **Forståelse:** At forstå og anvende OOP-principper kræver praktisk erfaring og kontekstuel anvendelse.
- **Design:** Effektivt design kræver omhyggelig overvejelse af hvilken abstraktion, arv og indkapsling der passer bedst til problemet.

Disse noter giver en oversigt over de grundlæggende koncepter inden for objektorienteret programmering, som behandles i kurset.