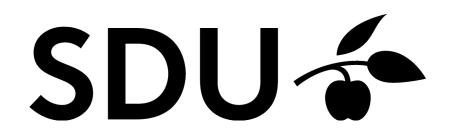
DM575

Forelæsning 04: Erklæring af klasser

Casper Bach Institut for Matematik og Datalogi



Klasser

- Objekter har attributter der beskriver deres indre tilstand, og metoder der beskriver deres opførsel.
 - **Eksempel:** Rektangler kan have forskellige sidelængder, så en Rectangle-klasse vil have to attributter der angiver disse. Rektangler har også et areal der f.eks. kunne beregnes ud fra de to sidelængder med en metode int area().
- I Java skal enhver (offentlig) klasse defineres i deres egen fil, der har samme navn som klassen.

00 Princip #1: Indkapsling

- Indkapslingsprincippet: Klasser skal ikke eksponere (eksterne) brugere for (interne) detaljer om egen implementering.
- Attributter beskriver objektets indre tilstand, og kan også have invarianter som vi til enhver tid vil sikre os ikke bliver brudt.
 - Eksempel: Et rektangel kan ikke have negative sidelængder.
- Nogle metoder er ikke lavet til eksternt brug (f.eks. fordi de omhandler implementeringsdetaljer, eller midlertidigt bryder invarianter).
- For at overholde indkapslingsprincippet bør disse attributter og metoder skjules så brugere ikke kan afhænge af dem.

Synlighed

- Vi kan sikre at indkapslingsprincippet overholdes ved at kontrollere hvilke metoder og attributter der kan tilgås af brugere.
- Dette gøres i Java ved hjælp af nøgleordene public og private.
 - Både attributter og metoder kan kvalificeres med public og private.
 - Offentlige klasser er tilgængelige udenfor pakken, private klasser (defineres ved at undlade at skrive public foran klassenavnet!) er kun synlige for klasser i samme pakke.
- Offentlige attributter bryder meget ofte indkapslingsprincippet.

Konstruktører

- Til at lave nye objekter bruges nøgleordet new, men hvordan sikrer vi, at attributter er udfyldt med meningsfyldte værdier?
 - Eksempel: Det giver ikke mening at lave et rektangel uden at angive de to sidelængder.
- Dette gøres ved at definere konstruktører, der er en særlig slags statiske metoder der bliver kaldt når et objekt bliver skabt.
- Konstruktører kan bruges til at sikre, at invarianter om objektets tilstand opretholdes f.eks. at sidelængder i et rektangel ikke kan være negative.
- Hvis vi f.eks. gerne vil have, at brugeren angiver sidelængderne når de laver et rektangel, definerer vi en konstruktør

```
public Rectangle (int x, int y)
```

En klasse kan have flere konstruktører.

UML klassediagrammer

- Til at illustrere en klasses grænseflade (uden at beskrive *hvordan* denne er implementeret) er et standardværktøj *UML klassediagrammer.*
- Vores løbende eksempel med rektangelklassen har UML klassediagrammet til højre.

Rectangle

- height: int
- length: int
- + Rectangle (int x, int y)
- + area (): int

Til VS Code!

- Fælles øvelse: Definer Board-klassen fra sidst således, at vores tidligere implementering af kryds-og-bolle virker.
 - Kommenter implementeringen med javadoc-annoteringer og generer dokumentation.
 - Brug assert til at angive kontrakter.