

# GRUPPE II IN1010 – UKE 3

Sivert Fjeldstad Madsen

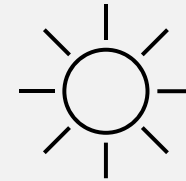
# EGENREFLEKSJON

- Åpne et tekstdokument eller finn noe å skrive på for hånd
- Bruk to minutter på å skrive ned alt du husker om arv og subklasser
- Fortsett å skrive helt til tiden er ute!



# I DAG

- Egenrefleksjon
- Arv
- Superklasser og subklasser
- Object
- Abstrakte klasser
- Klassehierarki
- Oppgaver



# ARV

- Et helt sentralt konsept innen objektorientering
- En måte å opprette objekter som deler enkelte egenskaper
- Kan gjøre at vi dupliserer mindre kode
- Kan gi oss en mer intuitiv måte å modellere data på



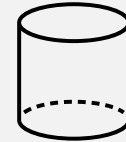
# SUPERKLASSER OG SUBKLASSER

- En superklasse er på en måte en classes «forelder»
- På samme måte er en subklasse på en måte en annen classes «barn»
- En superklasse defineres som en vanlig klasse
  - `class Superklasse {}`
- En subklasse defineres med nøkkelordet «extends»
  - `class Subklasse extends Superklasse {}`
- Subklassen vil ha alle instansvariablene og metodene som superklassen har



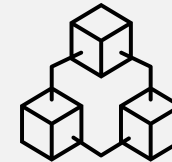
# KLASSEN OBJECT

- Man kan ha subklasser av subklasser!
  - Som et stort familietre
- Roten i Java sitt familietre er klassen «Object»
  - Alle klasser arver fra «class Object {}»
- Object har et par innebygde metoder som alle (sub)klasser også kan bruke
  - toString()
  - equals()
  - ...med flere!



# ABSTRAKTE KLASSER

- En abstrakt klasse er en klasse man ikke kan opprette en instans av
  - Gir oss en slags oppskrift for hvordan subclassene skal se ut
- Vi kan definere både instansvariabler og metoder i abstrakte klasser
- Syntaks:
  - **abstract** class Abstrakt {}
  - class Konkret extends Abstrakt {}



# PROTECTED

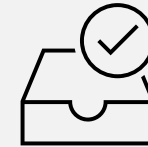
- Et tilgangsnivå mellom private og public
- Gjør variabelen eller metoden tilgjengelig for alle subklasser
  - ...men hindrer andre filer fra å aksessere dem
- En privat variabel eller metode i en superklasse vil ikke kunne aksessereres fra en subklasse!
  - Derfor bruker vi **protected**





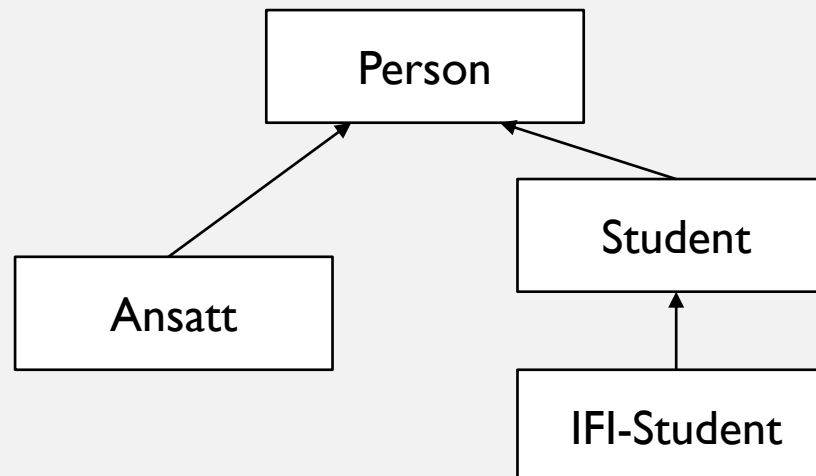
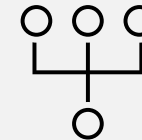
# INSTANCEOF

- Brukes for å sjekke om et objekt er av en gitt klasse
- Undersøker egentlig om objektet inneholder «egenskapene» til det du tester mot
- Gitt at man har klassene «Super» og «Sub extends Super» med instansene `Super sup = new Super();` og `Sub sub = new Sub;`, vil følgende være sant:
  - `sup instanceof Super;`
  - `sub instanceof Sub;`
  - `sub instanceof Super;`



# KLASSEHIERARKI

- En annen type tegning enn datastrukturtegning
- Viser klassenes «familietre»
- Bokser med klassenavn, hvor klassene peker «opp» på klassene de arver fra



# OPPSUMMERING

- Arv er et nytt verktøy til verktøykassen
- Kan gjøre mye gøy
- Utforsk!
  
- Ukas tips:
  - Versjonskontroll i VSCode
  - Publisert koden din enkelt til Github

