

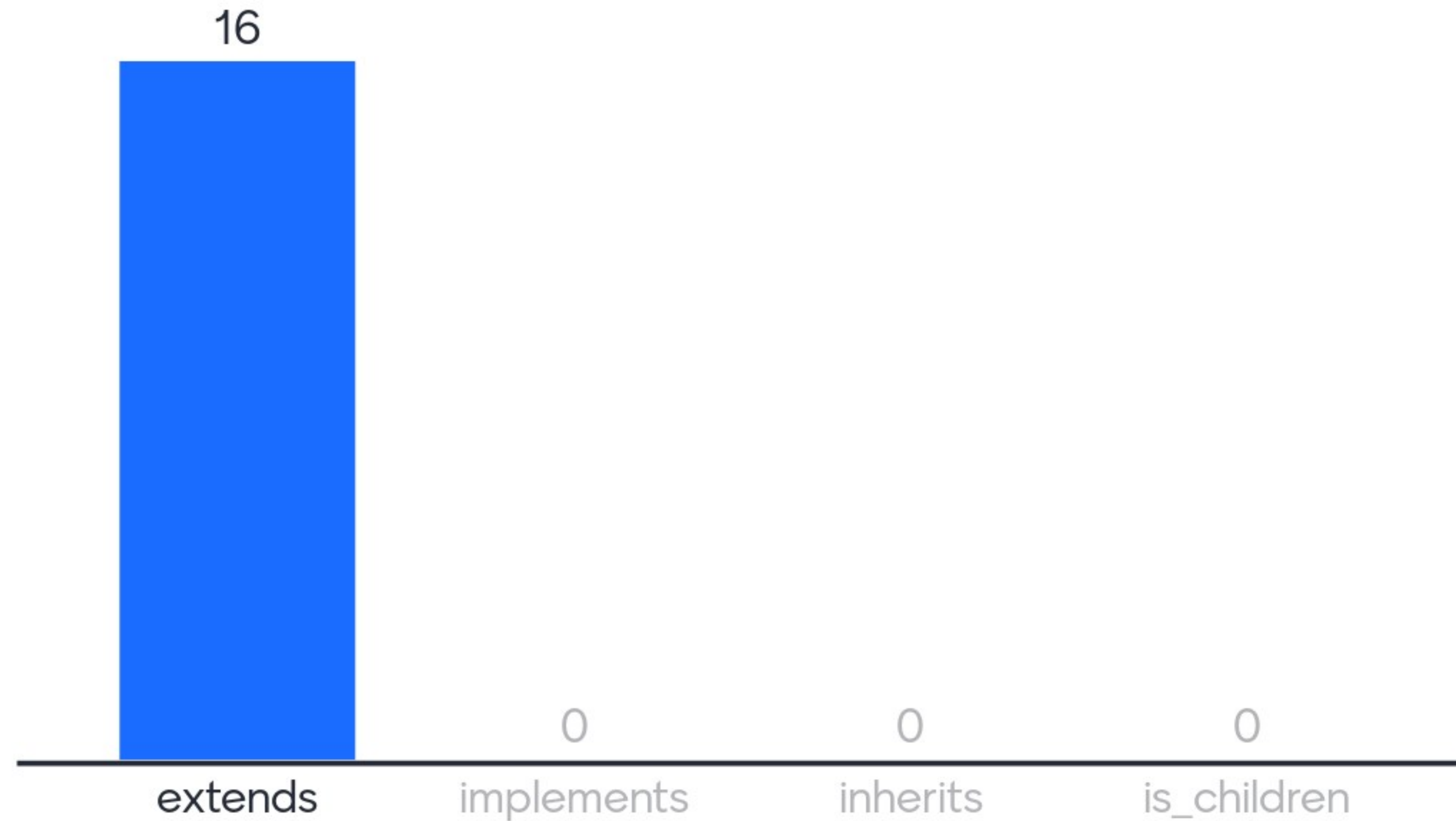
Seminartime - Uke 3

Arv og subklasser (fysisk gruppetime :))

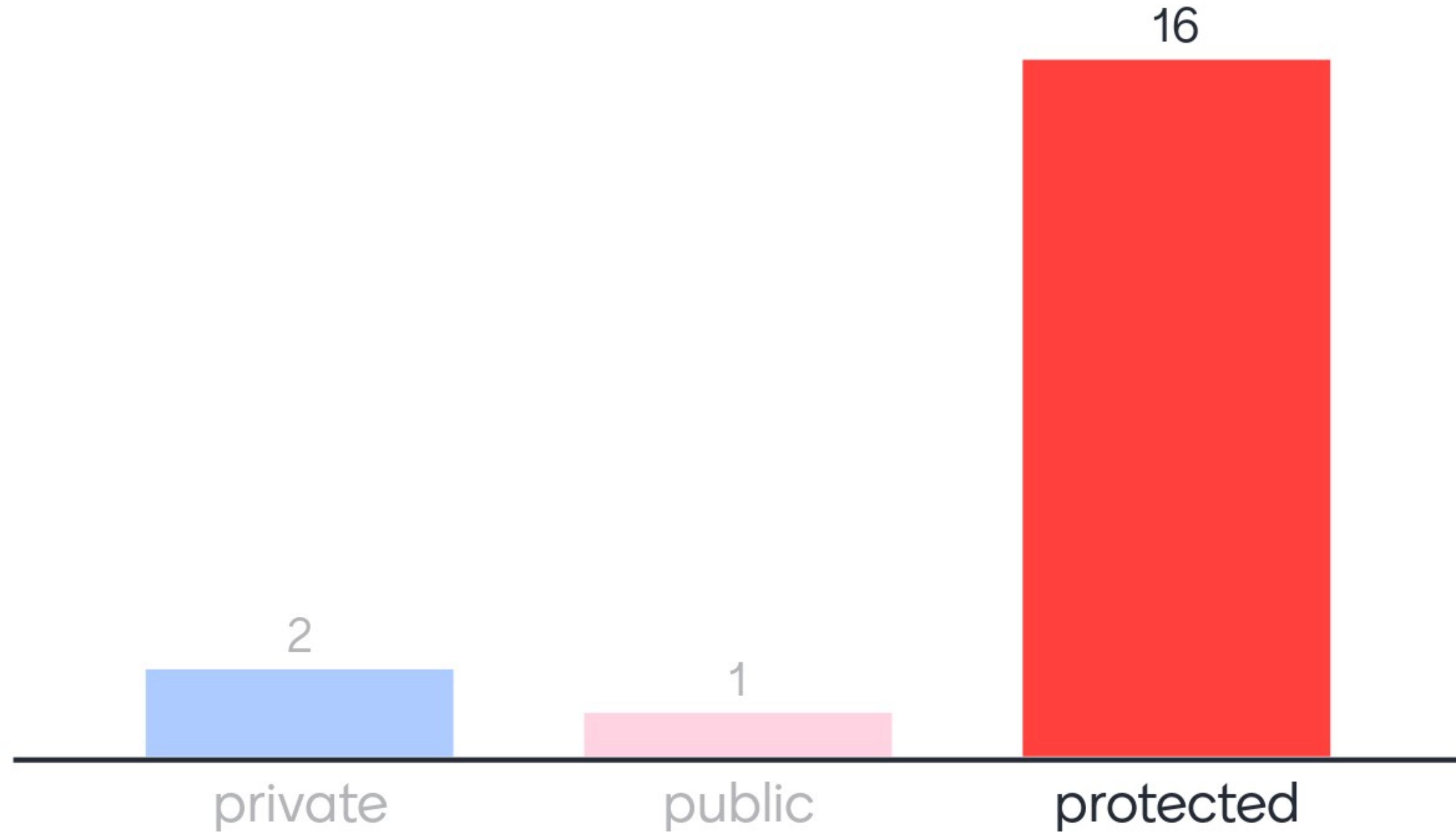
Agenda

- Spesialisering og generalisering
- Referanser, instanceof, protected
- Abstrakte klasser, casting, klassen Object
- Oppgaveløsning (evt. oblig)

... brukes for å indikere at en klasse arver fra en annen



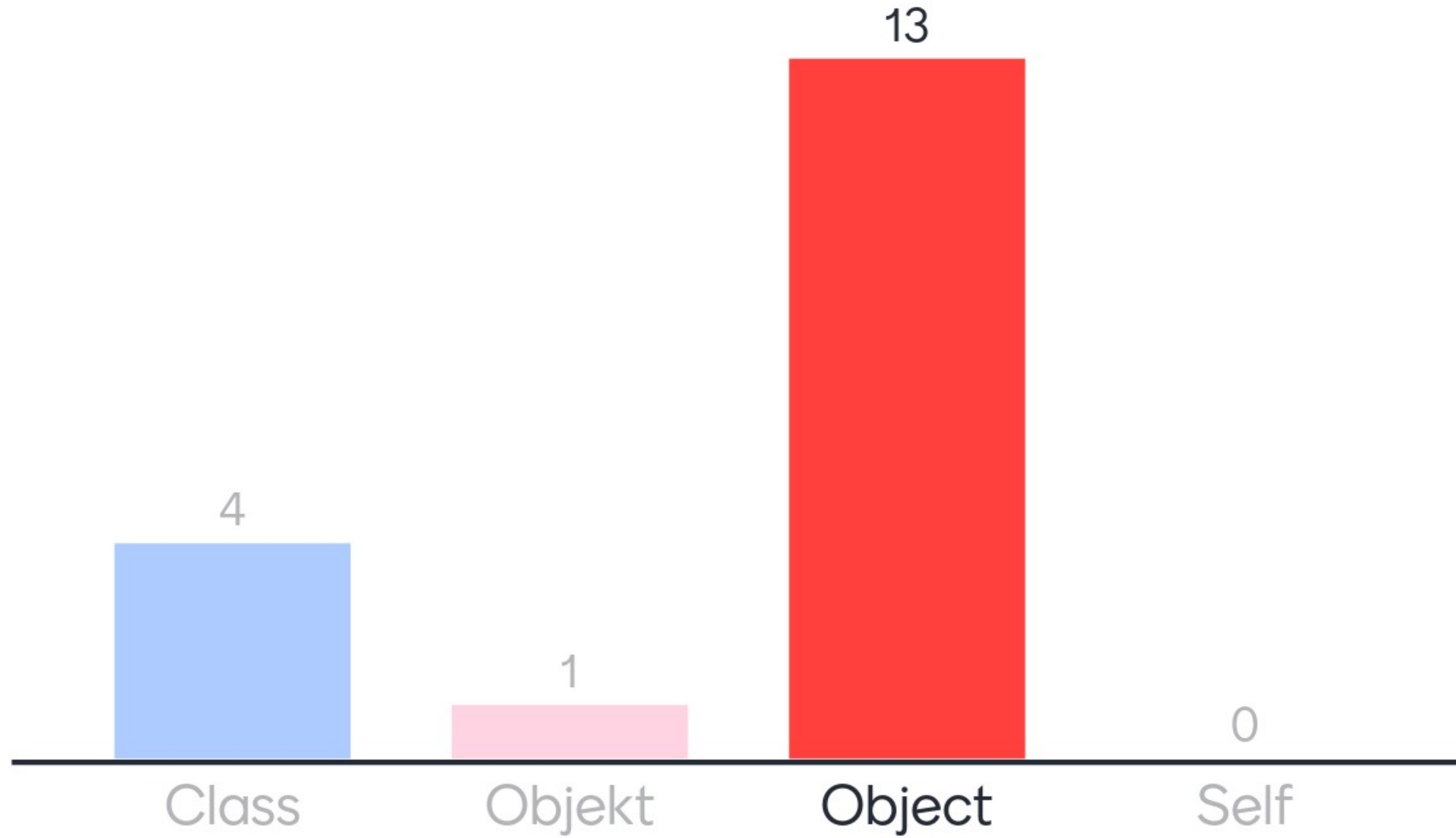
Gjør at kun subklasser kan se egenskapen



Et array i Java...



Alle klassers mor...



Arv og subklasser

- Lage oversiktlige og utvidbare komponenter og moduler
- Nøkkelord: extends
- Hierarkier av klasser (klassehierarki)
- En subklasse kan bruke metoder og variabler fra foreldreklassen

Spesialisering og generalisering

- En subklasse en er mer spesialisert utgave av foreldreklassen
- En Person-klasse er generell, mens Student er spesialisert
- Superklasser



Referanser

- Hvilke "briller" vi bruker for å se objektene
- Kan spille forskjellige roller
- Referanseverdiens type som avgjør hva vi har rett til!



Casting

- Lar systemet sjekke og konvertere (om det er mulig)
- Kalles også å typekonvertere
- Kan ikke caste lenger ned i hierarkiet enn et objekt tillater
- `ClassCastException`
- Nøkkelord: `instanceof`



Protected, Object-klassen, og abstrakte klasser

- Nøkkelord: protected
- Alle klassers mor: Object
- Konstanter - nøkkelord: final (konstanter bør være i store bokstave)
- Abstrakte klasser



Live-eksempel

Kjøretøy-klasser



Egenrefleksjon

Ett minutt på å skrive ned *alt* dere husker om arv og subklasser. Kun en regel; ikke lov å slutte å skrive.

Dette trenger ikke deles med andre, men fint å høre refleksjoner.



Oppgaveløsning m/ felles gjennomgang

→ [Oppgaver](#)

→ Kan også jobbe med oblig

→ Ikke nøl med å spørre om hjelp hvis dere lurer på noe

