

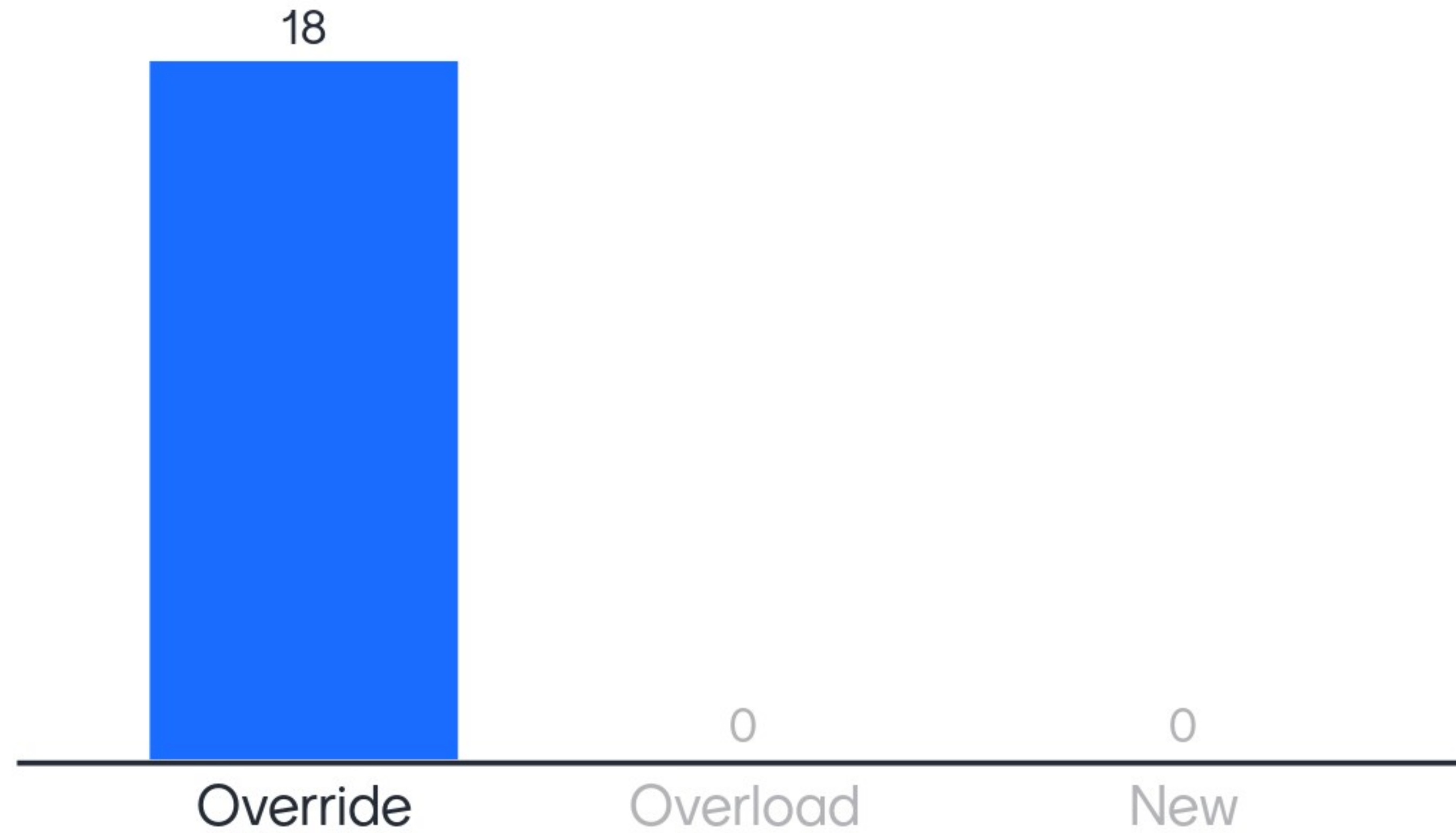
# IN1010 Seminarartime - Uke 4

Arv og polymorfi

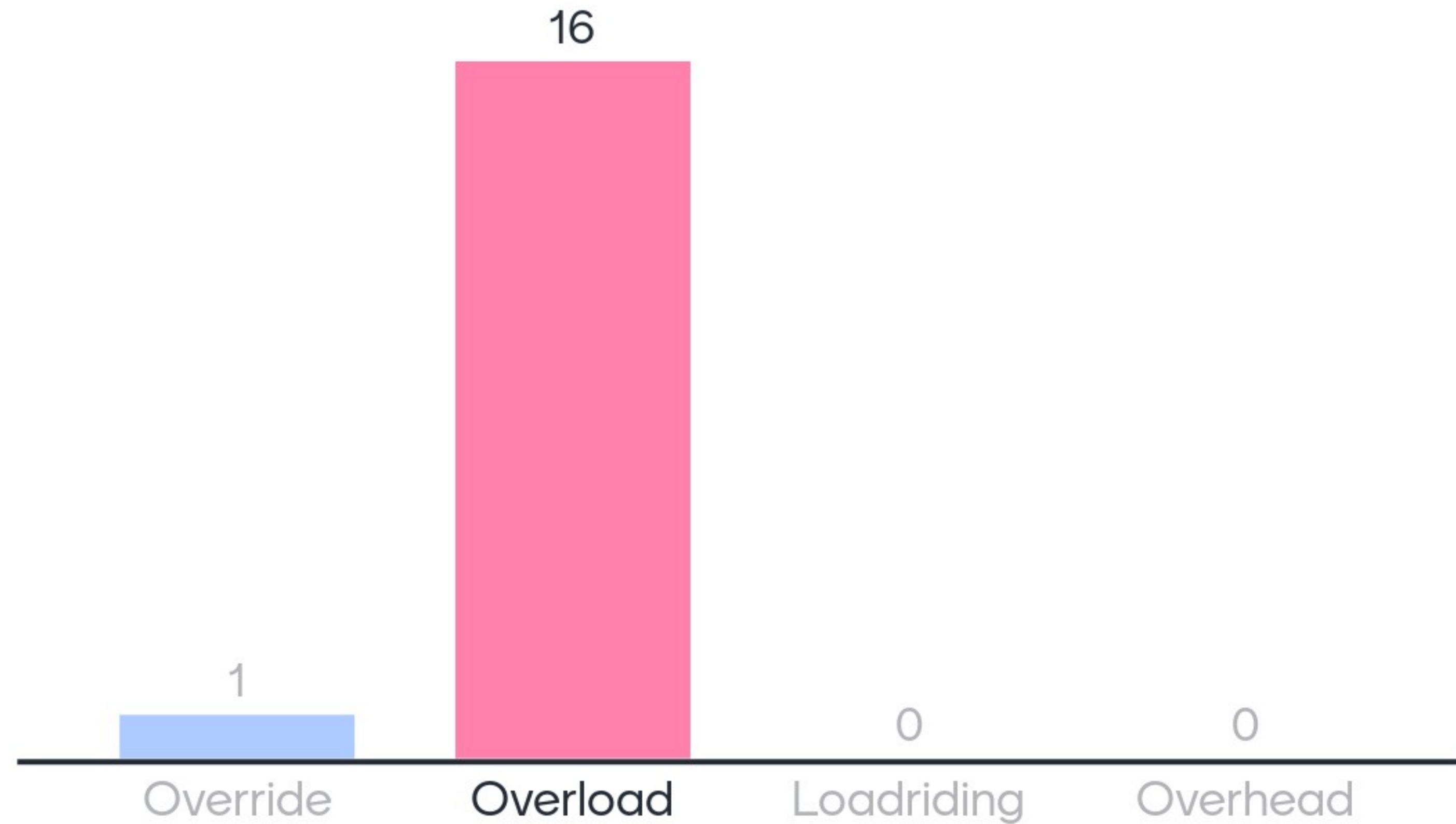
# Agenda

- Polymorfi
- Forskjellen på *overload* og *override*
- Object-klassen: `toString()` og `equals()`
- Konstruktører i superklassen

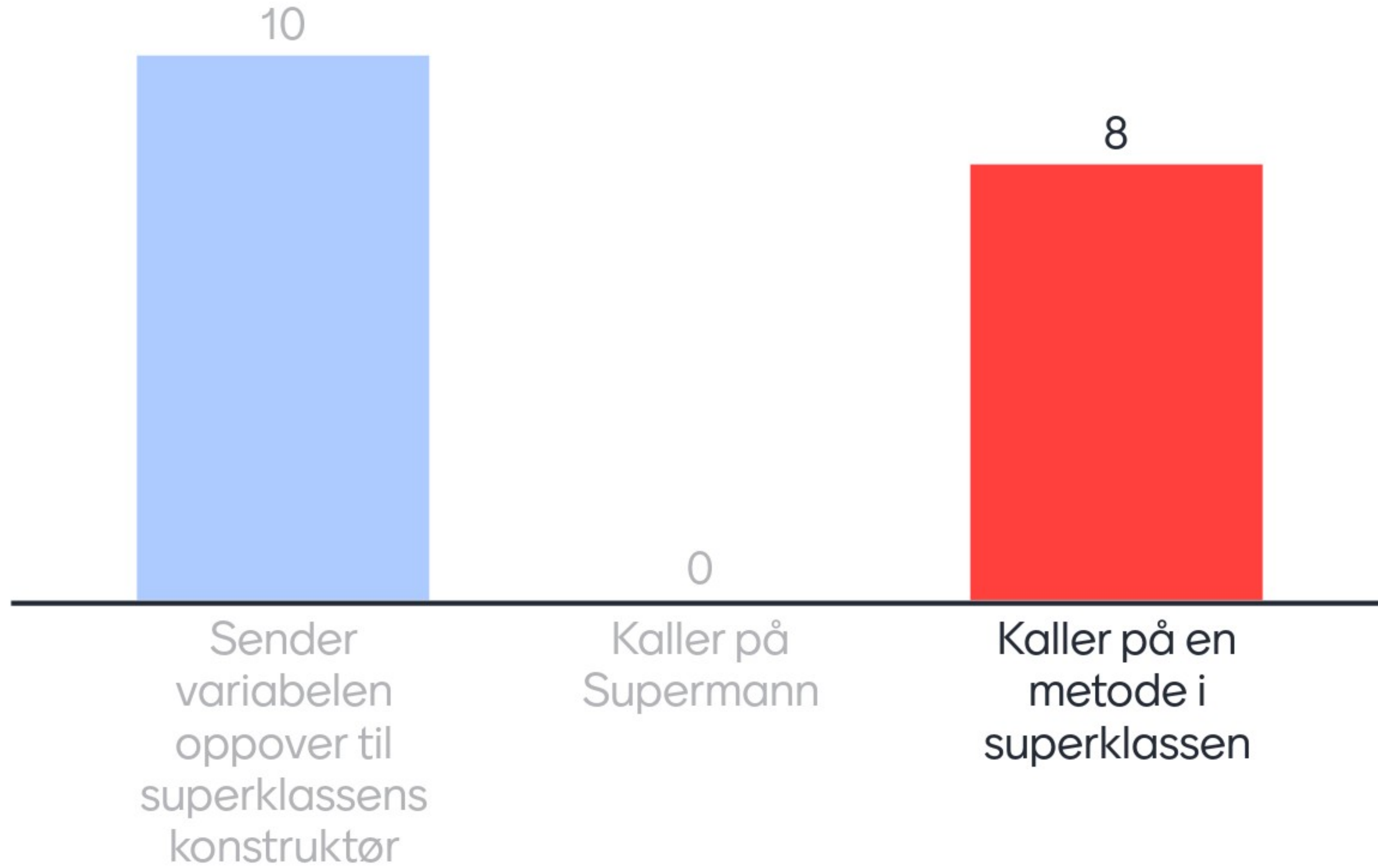
# For å redefinere eksisterende metoder bruker jeg...



# Å definere metoder med like navn men ulike signaturer kalles

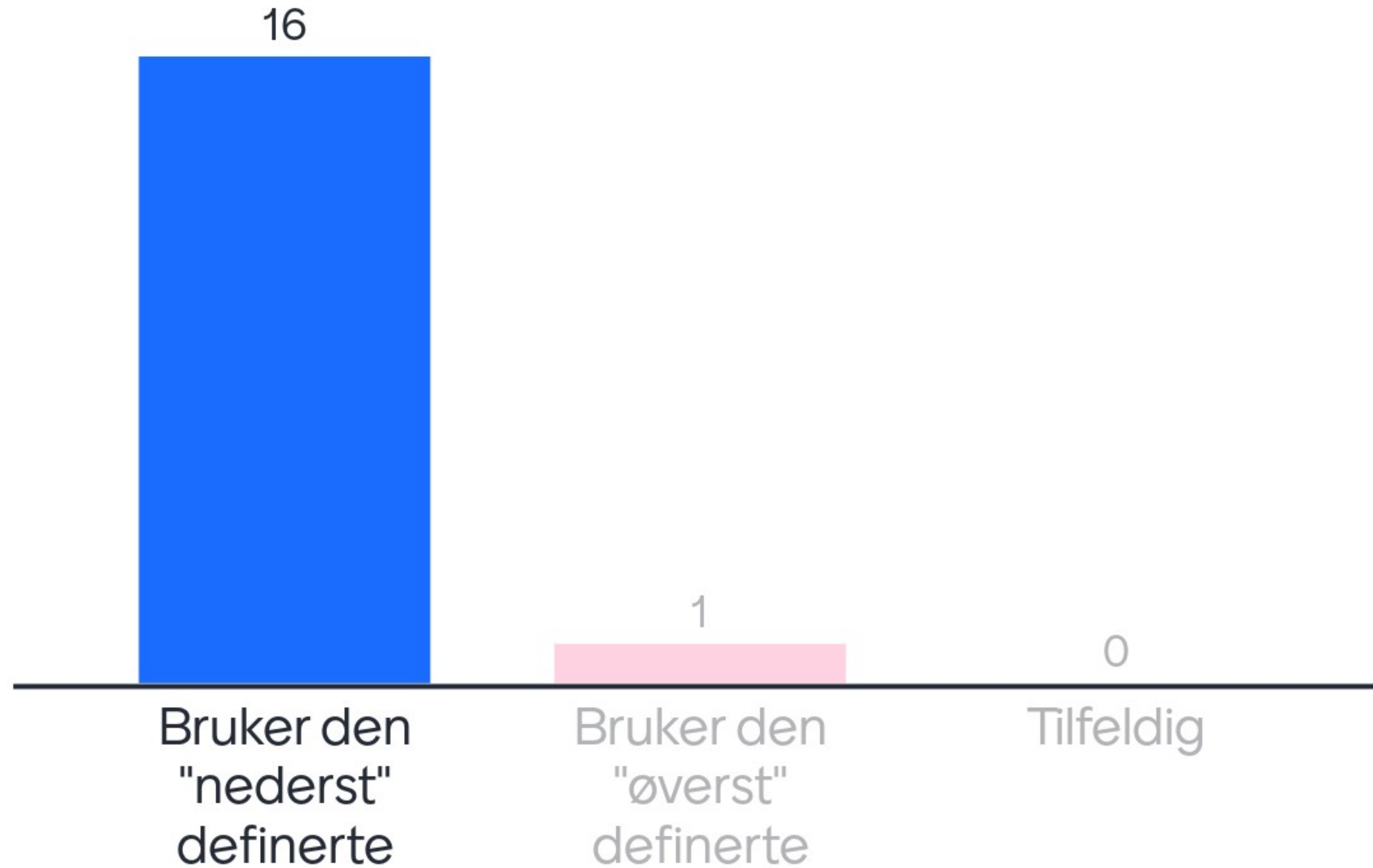


# Å skrive "super."

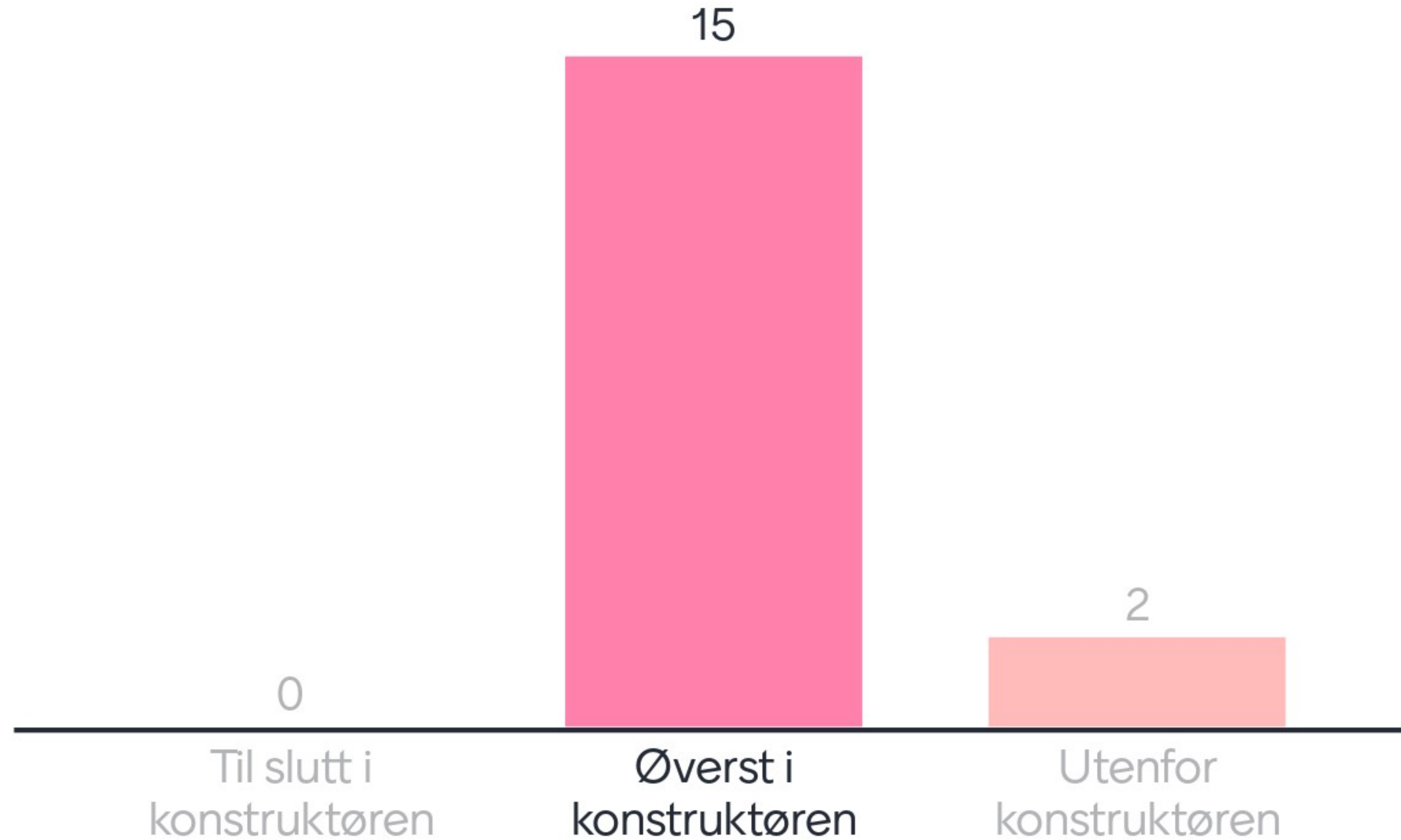




# Hvordan systemet vet hvilken metode den skal benytte seg av hvis det er flere polymorfe metoder



# Et kall på `super()` må legges...



# Polymorfi

- Klassene får ansvaret for implementering av metodene
- Brukeren kaller på metoden, objektet selv avgjør hva som skjer
- Bruker metoden i den "nederste" subklassen (???)
- Nøkkelord: override, annotering: @Override
- Men hva med instanceof?



# Overload vs. override

- Overloade er å definere flere metoder med like navn men ulike signaturer
- Overloade har ikke noe med polymorfi å gjøre, men er likvel nyttig
- Override overskriver tidligere eksisterende metoder
- Kan ikke override, dersom det ikke eksisterer fra før

# Livedemo av toString() og equals(), og konstruktører i superklassen

Mat-klasse med grønnsaker og andre subklasser av dette

# Hva nå?

→ Oppgaver

→ Lov å jobbe med oblig og andre ting, men oppgavene trener på konseptene