

IN1010 gruppe 1

Arv og subklasser

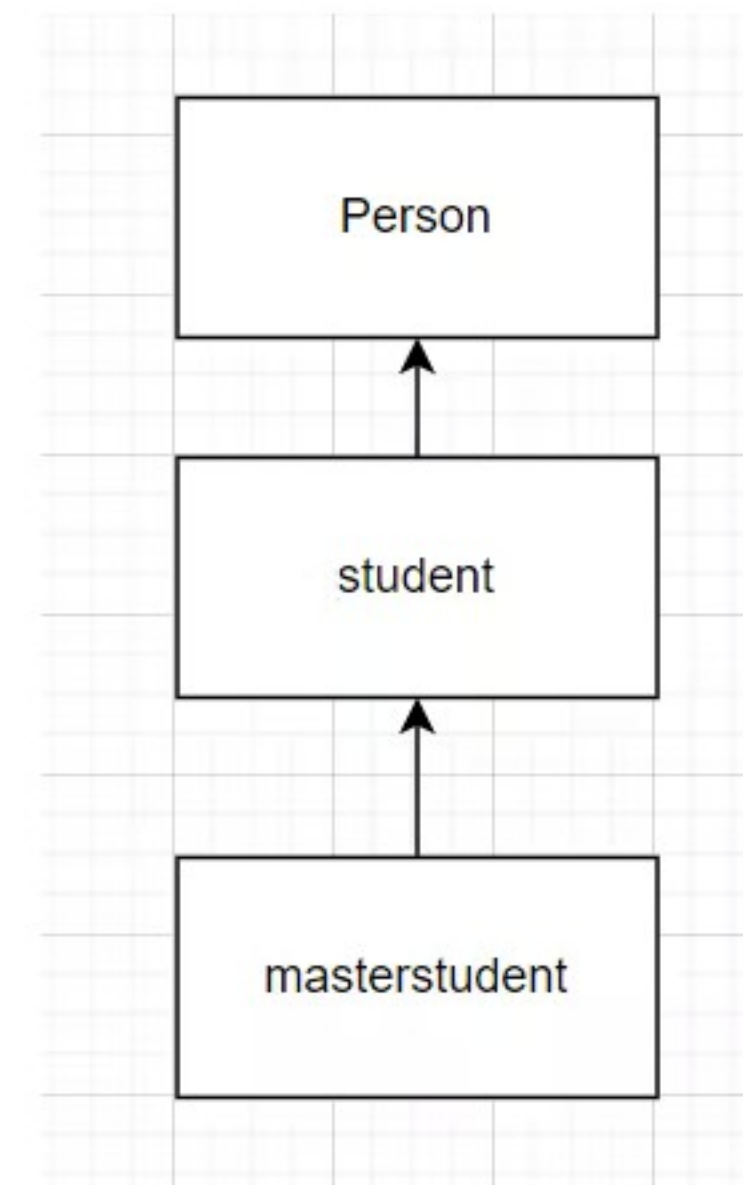


Opplegg for i dag

- Repetisjon
- quiz
- oppgaver

Klasser og subklasser

- En klasse beskriver et objekt med egenskaper
- En subklasse har de samme egenskapene som klassen den arver (superklassen) fra
 - en subklasse kan ha flere egenskaper
- extends angir om en klasse er subklasse. Eks Student extends Person
 - en student må være en person, men en person trenger ikke å være student
- Det går an å ha subklasse av en subklasse



Referansetyper

- En referanse til et objekt kan ha rolle
- for eksempel kan objekt av "class Student extends Person" ha rollen Student eller Person
 - dersom rolle er person er kun person egenskapene synlig
 - dersom rolle er student er både student og person egenskapene synlig
- en referanse av person har kun lov å å peke på objekter som har person-egenskaper (subklassene)
- Lov å peke nedover, IKKE oppover

Casting

- For å "bytte" briller og se på en objekt med andre briller
- eks `Person pers = new Student()`
- Vi vil endre referansetype fra `Person` til `Student`
- `Student stud = (Student) pers`

Eksempel

- `class A {}`
- `class B extends A {}`
- Variabel av type A kan referere til objekt B, men ikke omvendt
- `A variabel1 = new B() //lov`
- `B variabel2 = new A() //lov`
- `B variabel3 = (B) variabel1`



Protected, private og public

- Protected gjør at alle subklasser kan se denne egenskapen. Men ingen utenfor klassen
- Private gjør at ingen subklasser kan se denne egenskapen
- public er som før
- en konstant i Java deklarereres med "final" og kan ikke endres
 - `protected final int variabelnavn;`



Klasse Object

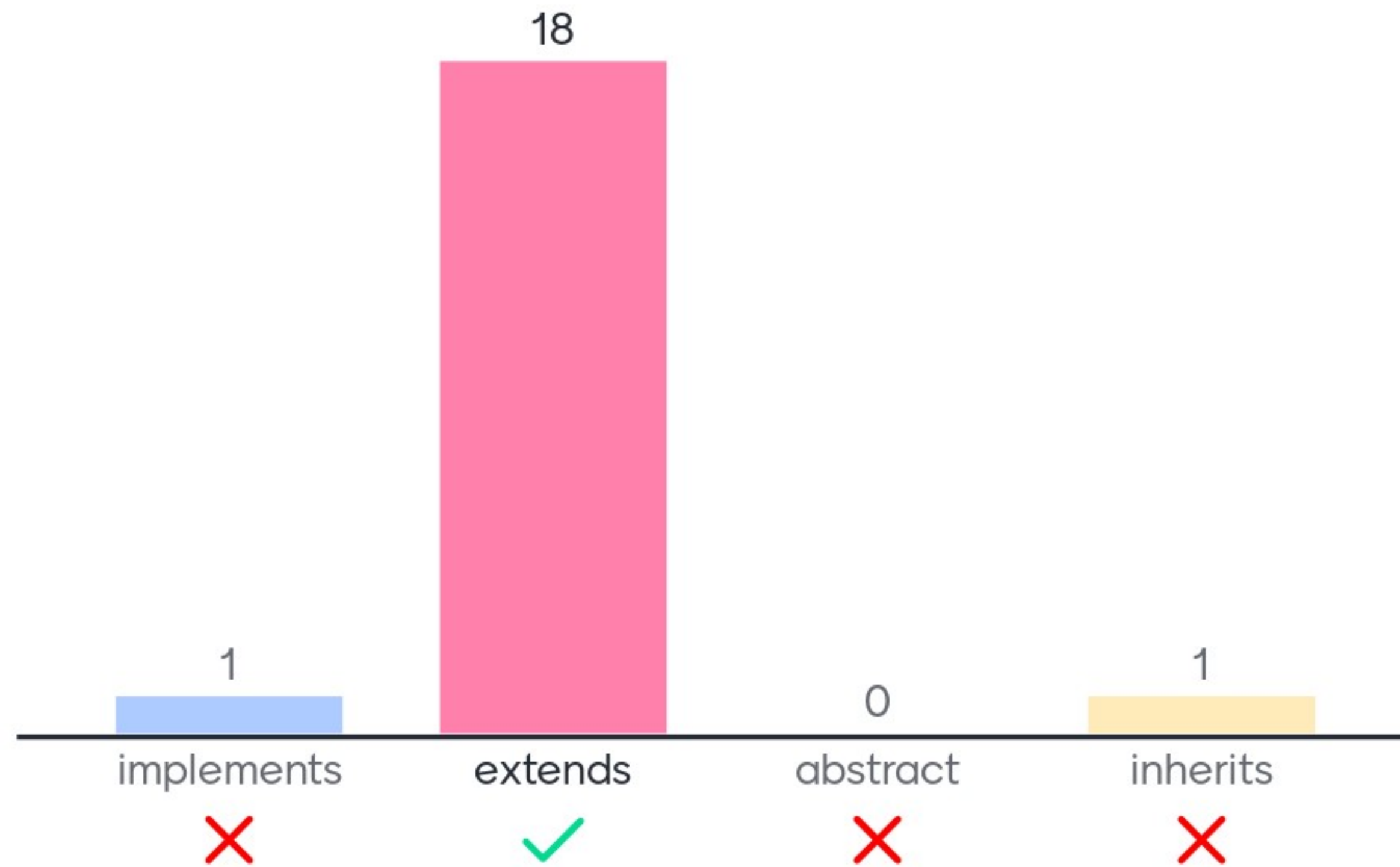
- er alle klassers superklasse
- `class Person {..}`
- er samme som `class Person extends Object {..}`



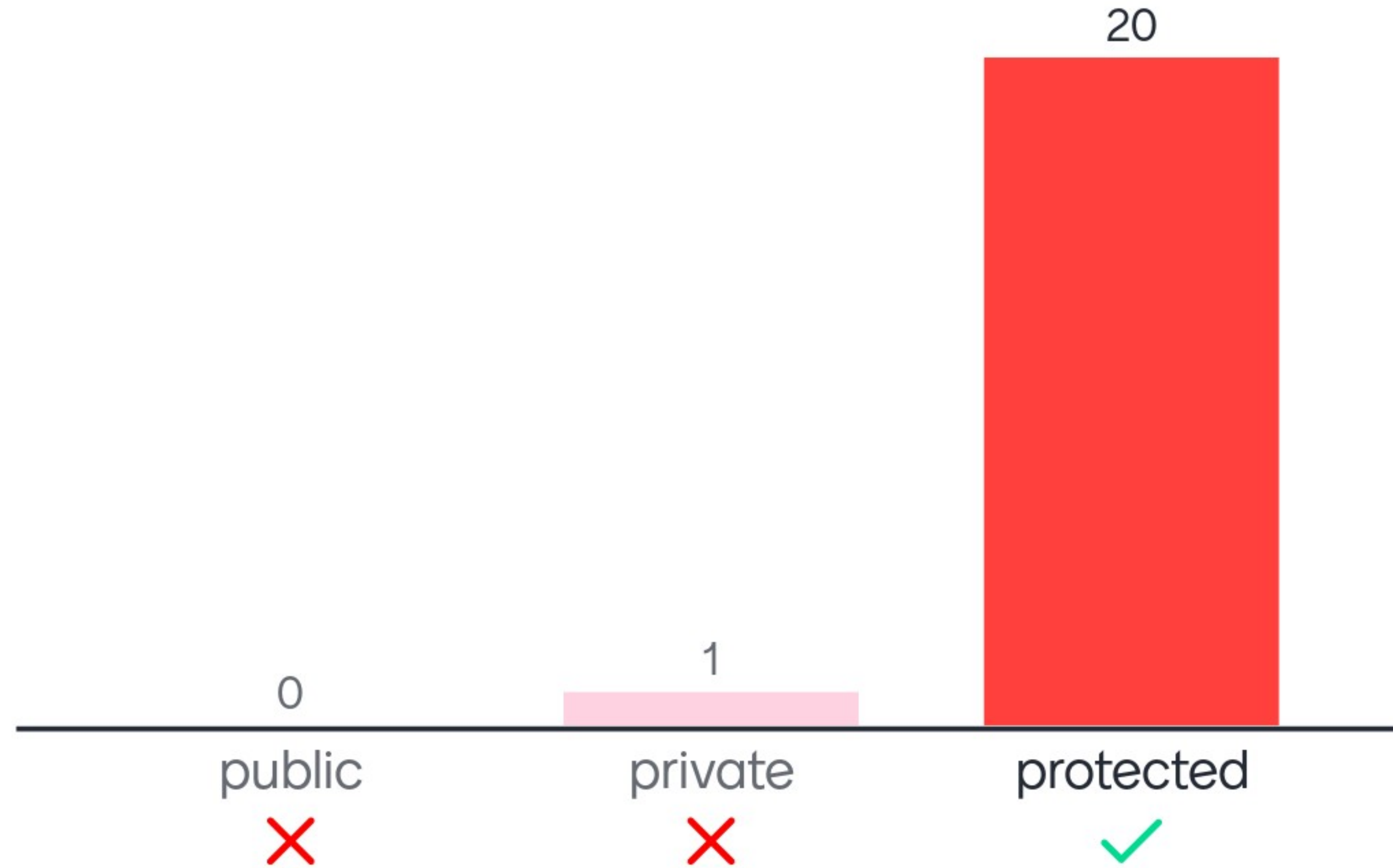
Abstrakte klasser

- går ikke an å lage objekter av abstrakte klasser
- går an å lage subklasser av abstrakte klasser som ikke er abstrakte og lage objekter av disse

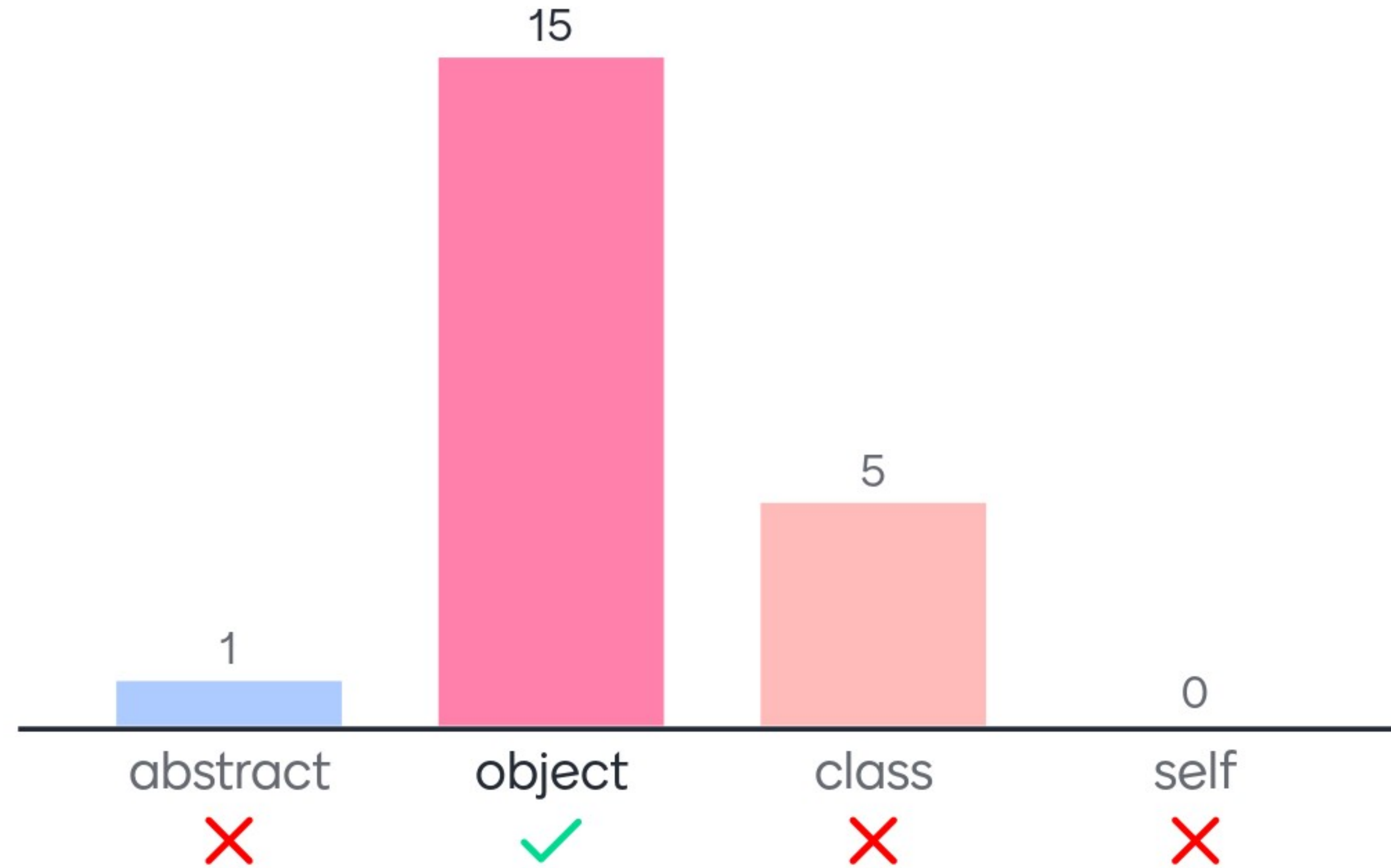
... brukes for å indikere at en klasse arver fra en annen



gjør at kun subklasser kan se variabelen

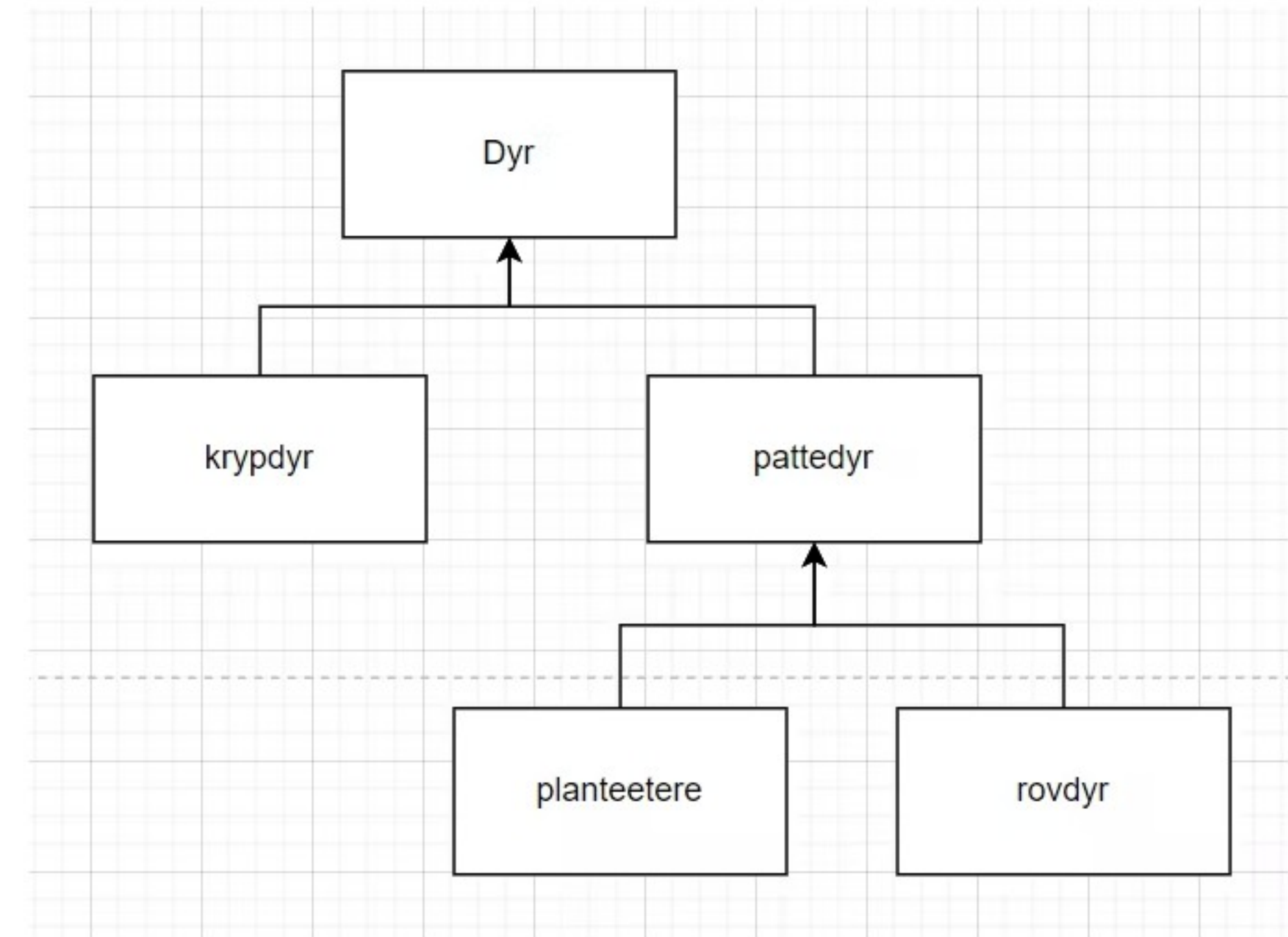


Alle klassers mor / superklasse



Eksempel dyr

- lag klassene og en metode som heter snakk
- lag et program som lager objekter av klassene



Oppgave egenrefleksjon

Alle studenter får ett minutt på å skrive ned alt de husker om arv og subklasser. Det er en regel, og det er at man ikke får lov slutte å skrive. (Dette skal ikke deles, skal bare hjelpe studenten med å komme i gang)



```
class A {}
class B extends A {}
class C extends A {}
class D extends C {}

class TrueOrFalse {
    public static void main(String[] args) {
        A a = new A();
        B b = new B();
        C c = new C();
        D d = new D();
        A e = new D();
        A f = new B();

        System.out.println("a) " + (b instanceof A));
        System.out.println("b) " + (c instanceof D));
        System.out.println("c) " + (d instanceof A));
        System.out.println("d) " + (e instanceof C));
        System.out.println("e) " + (f instanceof C));
    }
}
```

diskuter om utsagnene er true eller false



```
//Tilfelle 1
B ab = (B) a;

//Tilfelle 2
if( e instanceof B ){
    B eb = (B) e;
}

//Tilfelle 3
D ed = (D) e;

//Tilfelle 4
if( d instanceof D ){
    D ed = (D) e;
}

//Tilfelle 5
B fb = (B) f;

//Tilfelle 6
C ec = (C) e;
```

a) I hvert av tilfellene under diskuter om koden kommer eller ikke kommer til å kræsje, forklar hvorfor. b) Vil du si at tilfellet 3 eller 4 er best?

