

Seminaroppgaver uke 4

1. Dyr, Hynd Katt | Programmering, Diskusjon og Tegning

- Skriv en abstrakt klasse Dyr. Klassen skal være tom, utenom en abstrakt metode lagLyd(). Lag deretter to subclasser av dyr; Hund og Katt. Om man kaller på lagLyd i et hundobjekt skal det printes "Voff!" til terminalen. Om man kaller på lagLyd i et katteobjekt skal det printes "Mjau!"
- Diskuter:** Hvilken fordel får vi ved å ha en superklasse med en tom metode lagLyd? I felleskap/gruppen: Skriv et testprogram som demonstrerer denne fordelene.
- På egen hånd:** Tegn klassehierarki og datastruktur til oppgaven.

Diskuter: Sammenlign tegninger. Hva har dere gjort ulikt?

2. Personer og studenter | Diskusjon

a.

Person og student:

Diskuter hvilke metoder her som er eksempler på overriding/redefinering, og hvilke metoder som er eksempler på overloading/overlating. Fyll inn @Override der det hører hjemme.

Kode:

```
class Person {
    protected String navn = "ikke registrert.";
    protected String adresse = "ikke registrert.";

    public void registrerPerson(String n) {
        navn = n;
    }

    public void registrerPerson(String n, String a) {
        navn = n;
        adresse = a;
    }

    public void skrivPerson(){
        System.out.println("Navn: " + navn + ",
        adresse: " + adresse);
    }
}

class Student extends Person {
    protected String idnr = "ikke registrert.";

    public void registrerPerson(String n, String i) {
        navn = n;
        idnr = i;
    }

    public void registrerPerson(String n, String i, String
    a) {
        navn = n;
        idnr = i;
        adresse = a;
    }

    public void skrivPerson(){
        System.out.println("Navn: " + navn + ",
        studentnr: " + idnr + ", adresse: " + adresse);
    }
}
```

b. Diskusjon

Diskusjon1 : hva skrives ut her:

```
Person a = new Student();  
a.registrerPerson("Lise", "Holmveien 2");  
a.skrivPerson();
```

Diskusjon 2: Hva skrives ut her:

```
Person a = new Student();  
a.registrerPerson("Siri");  
a.skrivPerson();
```

Diskusjon 3: Hva skjer her:

```
Person a = new Person();  
a.registrerPerson("Siri", "1235", "Trimveien 9");  
a.skrivPerson();
```

3. Arv og subclasser | Programmering (Fortsettelse fra oppgave 4 fra uke 3)

Test metoden dine i et tesprogram mens du løser oppgaven

- a. Endringer i klassen Sjaafor:
 - i. Som vi vet, må en sjåfør være eldre enn 17 år (18 år og oppover). Hvis noen prøve å opprette en sjåfør som er yngre enn 18 skal det kastes en `IllegalArgumentException` med en beskjed om at en sjåfør ikke kan være yngre enn 18 år
 - ii. Endre konstruktøren slik at den sender navn og alder videre til superklassen sin (Person) **Hint:** super
- b. Endringer i klassen Person:
 - i. Person skal ha et navn og en alder ved opprettelse, hvis det ikke allerede er lagt inn fra forrige uke
 - ii. Lag en metode som henter navnet (kall denne hentNavn)
 - iii. Lag en metode som henter alderen (kall denne hentAlder)
- c. Endringer i klassen Kjøretøy:
 - i. Endre det slik at Kjøretøy klassen blir abstract. **Diskusjon:** Hvorfor ønsker vi det ?
 - ii. Endre slik at det er Kjøretøy klassen som tar vare på registreringsnummeret og sjåføren
 - iii. Lag en metode som henter registreringsnummer og en som henter sjåføren (hentRegNr og hentSjaafor)
- d. Endringer i klassen Motorsykel:
 - i. Endre slik at konstruktøren sender registreringsnummer og sjåføren videre til superklassen
- e. Endringer i klassen Bil
 - i. Endre slik at konstruktøren sender registreringsnummer og sjåføren videre til superklassen
 - ii. Lag en metode hentPassasjerer som returnerer arrayet med passasjerer

- iii. Lag en metode til kalt fyllBil som tar en Person og prøver å plasserer den på den første ledig plassen. Hvis personen får en plass skal true returneres, false ellers
- iv. Lag en metode fyllBil som tar inn et array med personer og prøver å plassere så mange som mulig (Det er kun lov å plassere en person der hvis ledig plass). Den skal returnere true hvis alle får en plass og false ellers. Denne metoden skal også plassere personene på de første plassene som er ledige.
- f. Klassen Parkeringshus, her skal det ikke legges til noe nye metoder, bare pass på at de har de riktig navnene hentKjoretoy og settKjoretoy
- g. Bruk test klassen gitt for å teste programmet ditt
 - i. Det er viktig at klassenavnen og metode navnen er riktig, hvis ikke vil programmet kræsje.

4. Arv og subclasser | Programmering

- a) Lag en klasse Mobel som tar inn: vekt, dybde og bredde i konstruktøren.
 - i) Lag metoder til å hente ut vekten.
 - ii) Lag en metode som henter ut hvor stort plass den tar (kall den hentStorrelse). Som er dybde * bredde
- b) Lag subclassen Sofa av Mobel, i tillegg til vekt, dybde og bredde tar en sofa også inn hvor mange personer som kan sitte i den.
 - i) Lag en metode som printer ut antall plasser sofaen har
- c) Lag en subklasse Kommode av Mobel, i tillegg til vekt, dybde, og bredde tar kommden også inn høyde, og antall skuffer.
 - i) Endre metoden hentStorrelse slik at for kommode så er det dybde * bredde * høyde.
 - ii) Lag en metode som printer antall skuffer
- d) Lag et hovedprogam som tester alle klassene dine, med de ulike metoden