

## Løsning oppgave 1 (programmeringen):

```
interface DrikkBar {
    void drikkGlass();
    void fyllOpp();
}

class Drikke implements DrikkBar {
    protected double max;
    protected double innhold;

    public Drikke(double max){
        this.max = max;
        this.innhold = max;
    }

    public void fyllOpp(){
        innhold = max;
    }

    public void drikkGlass(){
        if (innhold < 2) {
            innhold = 0;
        } else {
            innhold -= 2;
        }
    }

    public String toString(){
        return "Saa mye igjen av drikken: " + innhold;
    }
}

class Ol extends Drikke {

    public Ol(int max){
        super(max);
    }

    @Override
    public void drikkGlass(){
        if (innhold < 5) {
            innhold = 0;
        } else {
            innhold -= 5;
        }
    }
}
```

```

@Override
public String toString(){
    return "Saa mye igjen av oelen: " + innhold;
}

}

class TestInterface {
    public static void main(String[] args) {
        Drikke vann = new Drikke(10);
        vann.drikkGlass();
        System.out.println(vann);

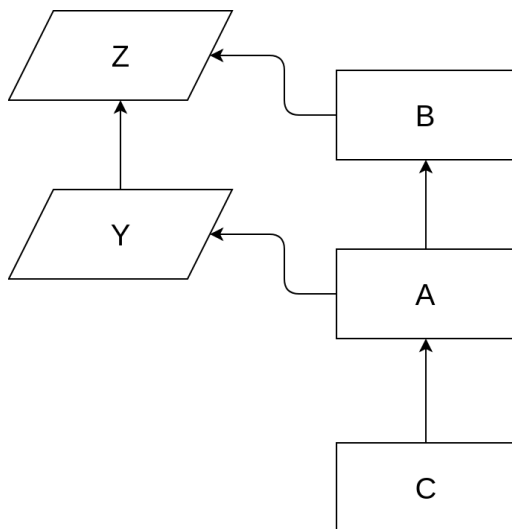
        Drikke ringnes = new Ol(40);
        ringnes.drikkGlass();
        System.out.println(ringnes);
    }
}

```

## 2.A

Gitt følgende klasse- og grensesnitt-definisjoner, tegn klassehierarkiet:

*Løsning:*



## 2.B

Diskusjon:

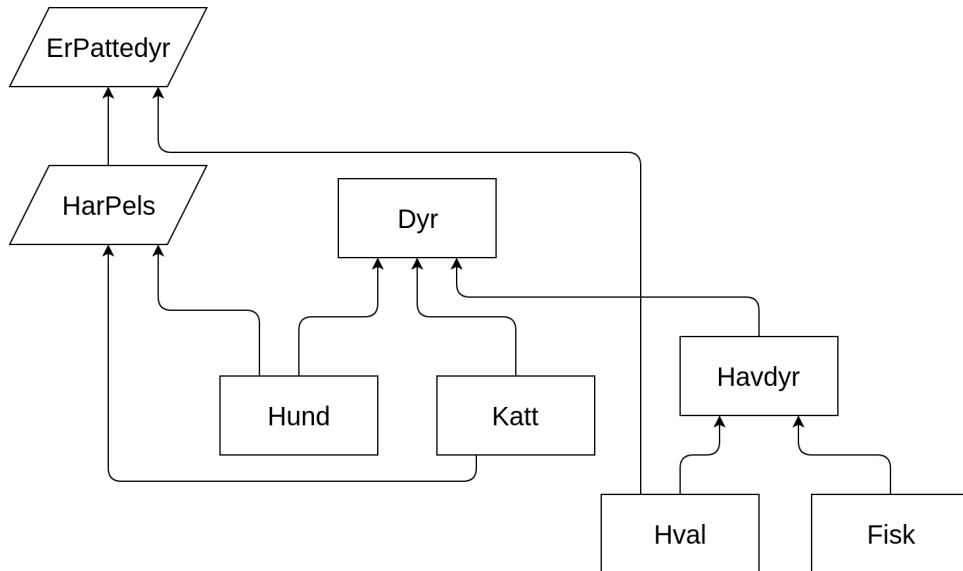
Hvilket, om noe, grensesnitt implementeres i C?

*Løsning:*

Siden C arver A vil det si at den også arver grensesnittet Y (som er en utvidelse av Z).

### 3.A

Gitt følgende klassehierarki, skriv opp klasse- og grensesnitt-definisjonene (som 2A).



### 4.A

Vi skal lage en enkel modell av en næringskjede. På toppen av næringskjeden finnes rovdynene og på bunnen finnes planteeterne. Skriv to grensesnitt, et for rovdyr og et for planteetere. Rovdyrene skal kunne jakte, mens planteeterne skal kunne beskytte seg selv.

### 4.B

Skriv klassene Bjorn, Ulv, Elg og Sau som implementerer de riktige grensesnittene over. Opprett objekter av klassene og legg alle rovdynene i en array av rovdyr, mens du legger alle planteeterne i en array av planteetere. Test programmet ditt.

*Diskusjon: hva gjør man med Bjørn, er den et rovdyr eller en planteeter, eller kan vi gjøre den til begge deler..?*

### 4.C

Tegn opp klassehierarkiet med alle klasser og grensesnitt.

### Løsning 4:

```
interface Planteeter {
    void beskytteSeg();
}
```

```

}

interface Rovdyr {
    void jakt();
}

class Sau implements Planteeter {
    public void beskytteSeg(){
        System.out.println("Sauen loper");
    }
}

class Elg implements Planteeter {
    public void beskytteSeg(){
        System.out.println("Elgen stanger angriperen");
    }
}

class Ulv implements Rovdyr {
    public void jakt(){
        System.out.println("Ulven jakter");
    }
}

class Bjorn implements Planteeter, Rovdyr{
    public void beskytteSeg(){
        System.out.println("Bjornen klorer angriperen");
    }

    public void jakt(){
        System.out.println("Bjornen jakter");
    }
}

class Dyrene {
    public static void main(String[] a){
        Rovdyr [] r = new Rovdyr[2];
        Planteeter [] p = new Planteeter[3];

        r[0] = new Ulv();
        r[1] = new Bjorn();

        p[0] = new Sau();
        Planteeter elg = new Elg();
    }
}

```

```
p[1] = elg;

p[2] = (Planteeter) r[1];

System.out.println("Alle Rovdyr:");
for(Rovdyr ro: r){
    ro.jakt();
}

System.out.println("Alle planteetere:");
for(int i = 0; i < p.length; i ++){
    p[i].beskytteSeg();
}
}
}
```