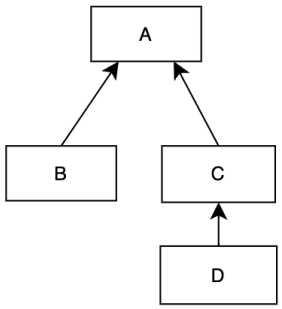
LØSNINGSFORSLAG - Seminaroppgaver, uke 3 - Arv og polymorfi II

2.) Instanceof og typekonvertering

For denne oppgaven er det til stor hjelp å tegne et klassehierarki for å holde orden på hvilke klasser som arver fra hvilke:



a) `b instanceof A` printer ut *true* hvis objektet som `b` peker på er av klassen `A` eller arver i fra `A`.
Utskriftene blir altså:

```
a) true
b) false
c) true
d) true
e) false
```

b)

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

Krasjer: a er av klassen A, som B arver fra. Vi kan derfor ikke caste a til B, siden den ikke har egenskapene til klasse B, bare til klasse A. (Generelt: Vi kan ikke caste et objekt av en superklasse til en subklasse)

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

Krasjer ikke: Hvis e er av klasse B eller arver i fra B, er det lov å caste e til B, og det er også dette vi sjekker i if-sjekken. Derimot blir ikke sjekken sann for dette tilfellet, siden e er av klasse D, som arver fra C, som arver fra A, ikke B.

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

Krasjer ikke: e er av klasse D, derfor har vi lov å caste e til D.

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

Krasjer ikke: e er av klasse D, derfor har vi lov å caste e til D. If-sjekken blir også sann her på grunn av dette.

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

Krasjer ikke: f er av klasse B, derfor har vi lov å caste f til B.

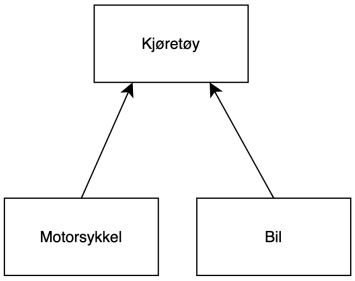
```
//Tilfelle 6
C ec = (C) e;
```

Krasjer ikke: e er av klasse D, som arver fra C. Derfor har vi lov å caste e til C.

c) Hvis vi ikke er 100% sikre på hvilken klasse e er av, er tilfelle 4 en sikrere løsning, siden den sjekker at e faktisk har egenskapene til klasse D før den caster e til D. Derfor vil tilfelle 4 aldri krasje, uansett hvilken klasse e er av, og er en bedre løsning.

3. Parkeringshus med motorsykler og biler

a) Klassehierarki



b) Løsningsforslag til programmet:

```java
public class TestParkeringshus {
    public static void main(String[] args) {
        Parkeringshus pHus = new Parkeringshus(5);
        pHus.settInn(new Bil("Bob", "abc123", 6), 0);
        pHus.settInn(new Bil("Anna", "aks827", 5), 0); // Tester full parkeringsplass
        pHus.settInn(new Motorsykkel("Tom", "xyz987"), 4);

        System.out.println("Skriver ut info");
        pHus.skrivInformasjon();

        pHus.taUt(0);

        System.out.println("Skriver ut info etter taUt(0)");
        pHus.skrivInformasjon();

    }
}

class Parkeringshus {
    int antPlasser;
    int iBruk = 0;
    Kjoeretoey[] parkeringsplasser;

    public Parkeringshus(int antPlasser) {
        this.antPlasser = antPlasser;
        this.parkeringsplasser = new Kjoeretoey[antPlasser];
    }

    public void settInn(Kjoeretoey kjoeretoey, int plass) {
        if (iBruk == antPlasser) {
            System.out.println("Parkeringshuset er fullt");
            return;
        }
        if (plass < 0 || plass >= antPlasser) {
            // Ugyldig plass
            throw new ArrayIndexOutOfBoundsException();
        }

        if (parkeringsplasser[plass] != null) {
            System.out.println("Parkeringsplassen er full");
            return;
        }

        parkeringsplasser[plass] = kjoeretoey;
        iBruk ++;
    }

    public Kjoeretoey taUt(int plass) {
        if (plass < 0 || plass >= antPlasser) {
            // Ugyldig plass
            throw new ArrayIndexOutOfBoundsException();
        }
        Kjoeretoey tarUt = parkeringsplasser[plass];
        parkeringsplasser[plass] = null;
        iBruk--;

        return tarUt;
    }

    // Skriver ut all informasjon om hvert av kjøretøyene
    public void skrivInformasjon() {
        for (Kjoeretoey kjoeretoey : parkeringsplasser) {
```

```
            if (kjoeretoey != null) {
                System.out.println(kjoeretoey + "\n");
            }
        }
    }
}


class Kjoeretoey {
    String sjaafoer;
    String regNr;

    public Kjoeretoey(String sjaafoer, String regNr) {
        this.sjaafoer = sjaafoer;
        this.regNr = regNr;
    }

    @Override
    public String toString() {
        return "Registreringsnummer:" + regNr + "\nTilhoerer: " + sjaafoer;
    }
}

class Bil extends Kjoeretoey {
    int antPersoner;

    public Bil(String sjaafoer, String regNr, int antPersoner) {
        super(sjaafoer, regNr);
        this.antPersoner = antPersoner;
    }

    @Override
    public String toString() {
        return "BIL\n" + super.toString() + "\nAntall personer: " + antPersoner;
    }
}

class Motorsykkel extends Kjoeretoey {

    public Motorsykkel(String sjaafoer, String regNr) {
        super(sjaafoer, regNr);
    }

    @Override
    public String toString() {
        return "MOTORSYKKEL\n" + super.toString();

    }
}
```