

Løsningsforslag – uke 3

(fra gjennomgang i gruppetimen)

1

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

2

- A kan peke på A
- B kan peke på B
- A kan peke på B

B kan ikke peke på A, fordi B er en subklasse av A.

3a

Mobel.java

```
public class Mobel{
    protected int vekt;
    protected double dybde;
    protected double bredde;

    public Mobel(int vekt, double dybde, double bredde){
        this.vekt = vekt;
        this.dybde = dybde;
        this.bredde = bredde;
    }

    public int hentVekt(){
        return vekt;
    }

    public double hentStorrelse(){
        return dybde * bredde;
    }
}
```

3b

Sofa.java

```
public class Sofa extends Mobel{
    private int antallPlasser;

    public Sofa(int vekt, double dybde, double bredde, int antPlasser){
        super(vekt, dybde, bredde);
        antallPlasser = antPlasser;
    }

    public void printAntallPlasser(){
        System.out.println("Sofaen har " + antallPlasser + " plasser");
    }
}
```

3c

Kommode.java

```
public class Kommode extends Mobil{
    private double hoyde;
    private int antallSkuffer;

    public Kommode(int vekt, double dybde, double bredde, double hoyde, int antSkuffer){
        super(vekt, dybde, bredde);
        this.hoyde = hoyde;
        antallSkuffer = antSkuffer;
    }

    @Override
    public double hentStorrelse(){
        return dybde*bredde*hoyde;
    }

    public void printAntallSkuffer(){
        System.out.println("Skuffen har " + antallSkuffer + " skuffer");
    }
}
```

3d

Hovedprogram.java

```
public class Hovedprogram{
    public static void main(String[] args) {
        Sofa sofa1 = new Sofa(89, 9.0, 45.6, 7);
        Kommode kommode1 = new Kommode(56, 2.5, 56.5, 34.6, 4);

        System.out.println("Storrelse paa sofaen: " + sofa1.hentStorrelse());
        System.out.println("Storrelse paa kommoden: " + kommode1.hentStorrelse());

        sofa1.printAntallPlasser();
        kommode1.printAntallSkuffer();
    }
}
```

4

Person.java

```
public class Person {
    private String navn;
    private int alder;

    public Person(String navn, int alder) {
        if ((alder < 0) || (alder > 120))
            throw new IllegalArgumentException("Ugyldig alder");
        this.navn = navn;
        this.alder = alder;
    }

    public String hentNavn() {
        return this.navn;
    }

    public int hentAlder() {
        return this.alder;
    }
}
```

Sjaafor.java

```
public class Sjaafor extends Person {
    public Sjaafor(String navn, int alder) {
        super(navn, alder);
        if (alder < 18)
            throw new IllegalArgumentException("Må være over 18 år!");
    }
}
```

Kjoretøy.java

```
public class Kjoretøy {
    private String regNr;
    protected int antSeter;
    protected Sjaafor sjaafor;

    public Kjoretøy(String regNr) {
        this.regNr = regNr;
    }

    public Kjoretøy(String regNr, Sjaafor sjaafor) {
        this.sjaafor = sjaafor;
    }

    public String hentRegNr() {
        return this.regNr;
    }
}
```

Bil.java

```
import java.util.List;

public class Bil extends Kjoretøy{
    private Person[] passasjerer;

    public Bil(String regNr, int antSeter) {
        super(regNr);
        this.antSeter = antSeter;
        this.passasjerer = new Person[antSeter-1];
    }

    public Bil(String regNr, Sjaafor sjaafor, int antSeter) {
        this(regNr, antSeter);
        this.sjaafor = sjaafor;
    }

    public void fyllBil(Sjaafor sjaafor, Person[] passasjerer) {
        if (sjaafor != null)
            settSjaafor(sjaafor);

        for (int i = 0; i < this.passasjerer.length; i++) {
            try {
                this.passasjerer[i] = passasjerer[i];
            }
            catch (ArrayIndexOutOfBoundsException e) {
                // Vil skje dersom passasjerer.length > this.antSeter.
            }
        }
    }
}
```

Motorsykkel.java

```
public class Motorsykkel extends Kjoretoy {
    public Motorsykkel(String regNr) {
        super(regNr);
        this.antSeter = 1;
    }

    public Motorsykkel(String regNr, Sjaafor sjaafor) {
        super(regNr);
        this.sjaafor = sjaafor;
    }
}
```

Parkeringshus.java

```
public class Parkeringshus {
    private int antPlasser;
    private Kjoretoy[] plasser;

    public Parkeringshus(int antPlasser) {
        this.antPlasser = antPlasser;
        this.plasser = new Kjoretoy[antPlasser];
    }

    public Kjoretoy hentKjoretoy(int plass) {
        try {
            return this.plasser[plass];
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Plassen du har valgt er ikke gyldig.");
            return null;
        }
    }

    public boolean settKjoretoy(int plass, Kjoretoy kjoretoy) {
        try {
            this.plasser[plass] = kjoretoy;
            return true;
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Plassen du har valgt er ugyldig.");
            return false;
        }
    }
}
```

Hovedprogram.java

```
public class Hovedprogram {
    public static void main(String[] args) {
        Person pappa = new Sjaafor("Pappa", 32);
        Person[] familie = {
            new Person("Mamma", 30),
            new Person("Henrik", 6),
            new Person("Albert", 5),
            new Person("Emma", 9)
        };
        Kjoretoy familiebil = new Bil("LY50102", 5);
        Motorsykkel harley = new Motorsykkel("BK5012", (Sjaafor) pappa);
        Parkeringshus qPark = new Parkeringshus(120);

        ((Bil)familiebil).fyllBil((Sjaafor) pappa, familie);

        qPark.settKjoretoy(1, familiebil);
    }
}
```