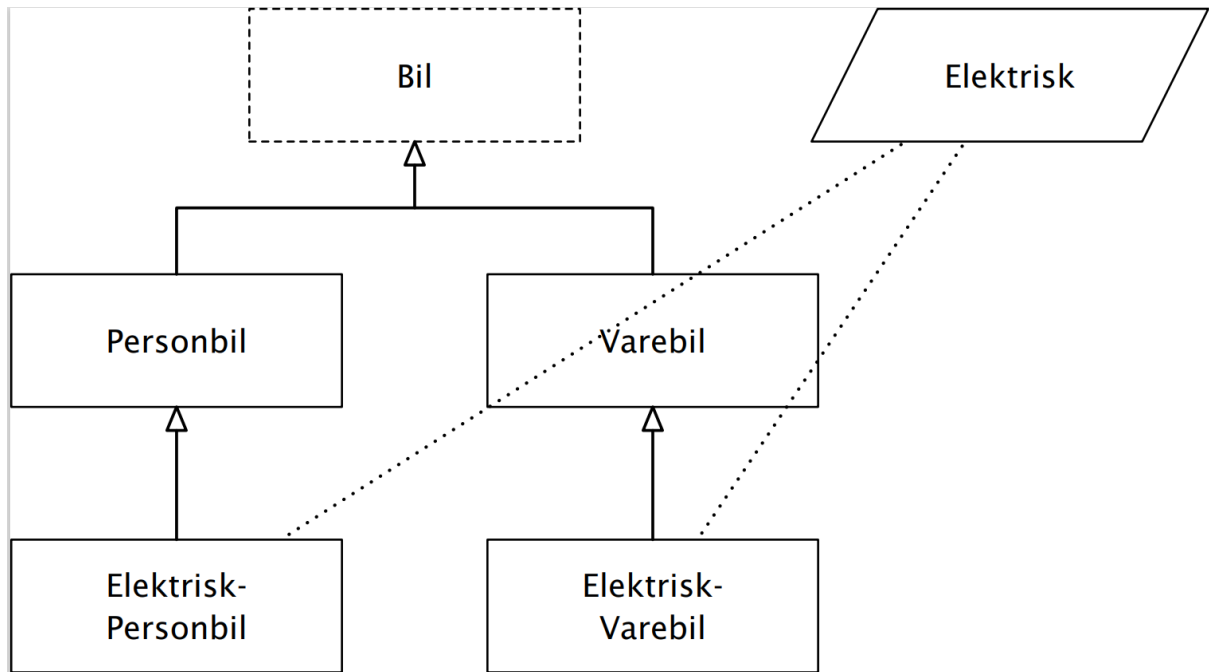
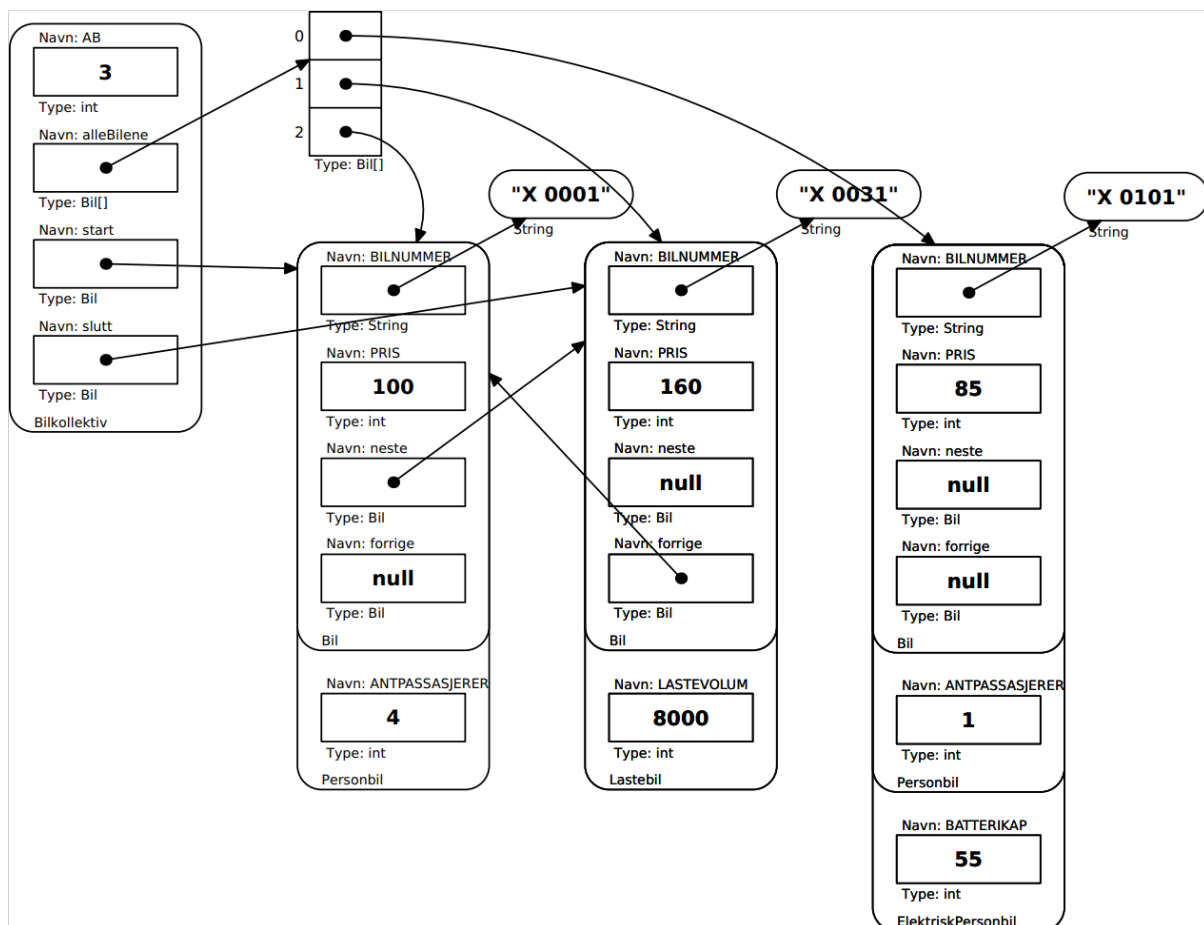


Løsningsforslag Eksamen 2022 Bilkollektiv

Oppgave 1



Oppgave 4



Kode

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.Scanner;

abstract class Bil {
    final String BILNUMMER;
    final int PRIS;
    Bil neste = null, forrige = null; // Listepekere

    Bil (String nr, int kr) {
        BILNUMMER = nr; PRIS = kr;
    }

    boolean erElbil () {
        return false;
    }
}

```

```

    Bil finnBilR (Dialog dialog, boolean kunElektrisk) {
        if (! kunElektrisk && dialog.svarJaEllerNei("Liker du " + this + "?"))
            return this;

        if (neste != null)
            return neste.finnBilR(dialog, kunElektrisk);
        return null;
    }

    @Override
    public String toString () {
        return BILNUMMER + " pris " + PRIS;
    }
}

class Personbil extends Bil {
    final int ANT_PASSASJERER;

    Personbil (String nr, int kr, int ant) {
        super(nr, kr);
        ANT_PASSASJERER = ant;
    }

    @Override
    public String toString () {
        return "Personbil " + super.toString() + " " + ANT_PASSASJERER + " pas";
    }
}

class Varebil extends Bil {
    final int LASTEVOLUM;

    Varebil (String nr, int kr, int vol) {
        super(nr, kr);
        LASTEVOLUM = vol;
    }

    @Override
    public String toString () {
        return "Varebil " + super.toString() + " " + LASTEVOLUM + " volum";
    }
}

interface Elektrisk {
    int hentBatterikapasitet ();
}

```

```

class ElektriskPersonbil extends Personbil implements Elektrisk {
    final int BATTERIKAP;

    ElektriskPersonbil (String nr, int kr, int vol, int kap) {
        super(nr, kr, vol);
        BATTERIKAP = kap;
    }

    @Override
    boolean erElbil () {
        return true;
    }

    @Override
    Bil finnBilR (Dialog dialog, boolean kunElektrisk) {
        if (dialog.svarJaEllerNei("Liker du " + this + "?"))
            return this;

        if (neste != null)
            return neste.finnBilR(dialog, kunElektrisk);
        return null;
    }

    @Override
    public int hentBatterikapasitet () {
        return BATTERIKAP;
    }

    @Override
    public String toString () {
        return super.toString() + " " + BATTERIKAP + " Kwh";
    }
}

class ElektriskVarebil extends Varebil implements Elektrisk {
    final int BATTERIKAP;

    ElektriskVarebil (String nr, int kr, int vol, int kap) {
        super(nr, kr, vol);
        BATTERIKAP = kap;
    }

    @Override
    boolean erElbil () {
        return true;
    }

    @Override

```

```

    Bil finnBilR (Dialog dialog, boolean kunElektrisk) {
        if (dialog.svarJaEllerNei("Liker du " + this + "?"))
            return this;

        if (neste != null)
            return neste.finnBilR(dialog, kunElektrisk);
        return null;
    }

    @Override
    public int hentBatterikapasitet () {
        return BATTERIKAP;
    }

    @Override
    public String toString () {
        return super.toString() + " " + BATTERIKAP + " Kwh";
    }
}

interface Dialog {
    boolean svarJaEllerNei (String sporsmal);
}

class TastaturDialog implements Dialog {
    Scanner tastatur = new Scanner(System.in);

    @Override
    public boolean svarJaEllerNei (String sporsmal) {
        while (true) {
            System.out.print(sporsmal + " ");
            String svar = tastatur.nextLine().trim().toLowerCase();
            if (svar.charAt(0) == 'j') return true;
            if (svar.charAt(0) == 'n') return false;
        }
    }
}

class GUIDialog implements Dialog {
    JFrame vindu = null;
    JPanel panel;
    JLabel tekstfelt;
    JButton jaknapp, neiknapp;

    Thread hovedtrad = Thread.currentThread();
    boolean svaret = true;

    @Override

```

```

public boolean svarJaEllerNei (String sporsmal) {
if (vindu == null) {
    try {
        UIManager.setLookAndFeel(
            UIManager.getCrossPlatformLookAndFeelClassName());
    } catch (Exception e) { System.exit(1); }
    vindu = new JFrame("JA eller NEI?");
    vindu.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    panel = new JPanel();
    vindu.add(panel);

    tekstfelt= new JLabel(sporsmal);
    panel.add(tekstfelt);

    class SvarJaNei implements ActionListener {
        boolean svar;

        SvarJaNei (boolean jn) {
            svar = jn;
        }

        @Override
        public void actionPerformed (ActionEvent e) {
            svaret = svar;
            hovedtrad.interrupt();
        }
    }

    jaknapp = new JButton("JA");
    jaknapp.addActionListener(new SvarJaNei(true));
    panel.add(jaknapp);

    neiknapp = new JButton("NEI");
    neiknapp.addActionListener(new SvarJaNei(false));
    panel.add(neiknapp);

    vindu.pack(); vindu.setVisible(true);
} else {
    tekstfelt.setText(sporsmal);
}

try {
    Thread.sleep(1000000);
} catch (InterruptedException e) {}

return svaret;
}

```

```

}

class Bilkollektiv {
    final int AB;
    Bil[] alleBilene;
    Bil start, slutt; // Liste av ledige biler

    Bilkollektiv (int ant) {
        AB = ant;
        alleBilene = new Bil[AB];
        start = slutt = null;
    }

    void lagBilPris () {
        int minForrigePris = -1;
        for (int n = 1; n <= AB; ++n) {
            // Finn den billigste bilen blant de som er igjen:
            Bil billigst = null;
            for (int i = 0; i < AB; ++i) {
                Bil b = alleBilene[i];
                if (b.PRIS > minForrigePris && (billigst == null || b.PRIS < billigst.PRIS))
                    billigst = b;
            }

            // Sett bilen inn sist i listen:
            if (start == null) {
                start = slutt = billigst;
            } else {
                slutt.neste = billigst;
                billigst.forrige = slutt;
                slutt = billigst;
            }
            minForrigePris = billigst.PRIS;
        }
    }

    void visBilene () {
        /* Til testing (ikke bedt om i oppgaven) */
        System.out.println("Test: Bilene er (sortert):");
        Bil b = start;
        while (b != null) {
            System.out.println("    " + b);
            b = b.neste;
        }
        System.out.println();
    }

    void taUtBil (Bil b) {

```

```

if (b==start && start==slutt) {
    start = slutt = null;
} else if (b == start) {
    start = start.neste;
    start.forrige = null;
} else if (b == slutt) {
    slutt = slutt.forrige;
    slutt.neste = null;
} else {
    b.forrige.neste = b.neste;
    b.neste.forrige = b.forrige;
}
b.neste = b.forrige = null;
visBilene();
}

Bil velgBil (Dialog d) {
boolean kunElbil = d.svarJaEllerNei("Er du bare interessert i elbil?");
Bil b = start;
while (b != null) {
    if (b.erElbil() || ! kunElbil) {
        if (d.svarJaEllerNei("Liker du " + b + "?")) {
            taUtBil(b);
            return b;
        }
    }
    b = b.neste;
}
return null;
}

Bil velgBilR (Dialog d) {
boolean kunElbil = d.svarJaEllerNei("Er du bare interessert i elbil?");
Bil b = start.finnBilR(d, kunElbil);
if (b != null)
    taUtBil(b);
return b;
}

/* Hovedprogram (ikke bedt om i oppgaven) */
public static void main (String[] arg) {
Bilkollektiv kol = new Bilkollektiv(3);
kol.alleBilene[0] = new Personbil("AA00001", 350, 4);
kol.alleBilene[1] = new ElektriskVarebil("AA00002", 745, 21, 50);
kol.alleBilene[2] = new ElektriskPersonbil("AA00003", 310, 3, 45);
kol.lagBilPris();
kol.visBilene();
}

```



```
Dialog d = new TastaturDialog();
// Dialog d = new GUIDialog();
for (int i = 1; i <= 3; ++i) {
    // Bil b = kol.velgBil(d);
    Bil b = kol.velgBilR(d);
    if (b == null)
        System.out.println("Ingen bil passet.");
    else
        System.out.println("Bil nr " + i + " er " + b + ".");
    kol.visBilene();
}
System.exit(0);
}
```