Norges Informasjonsteknologiske Høgskole

PG2100 – Programmering 2

Dato: 3.6.14

Oppgave 1

a)
```
public GroceryItem(String name, int quantity, double pricePerUnit) {
  setName(name);
  setQuantity(quantity);
  setPricePerUnit(pricePerUnit);
}
```

b)
```
public String toString() {
  return getQuantity() + " "
           + getName()
           + "(" + getPricePerUnit()
           + " per/stk)";
}
```

c)
```
public boolean equals(Object o) {
  if (!(o instanceof GroceryItem)) return false;
  if (o == this) return true;
  GroceryItem g = (GroceryItem) o;
  return getName().equals(g.getName()); //eller sammenligning på alle attributter
}
```

Alternativ 1:

d)
```
import java.util.ArrayList;

public class GroceryList {
  private ArrayList<GroceryItem> groceryList;

  public GroceryList() {
    groceryList = new ArrayList<GroceryItem>();
  }

  public boolean addItem(GroceryItem item) {
    return groceryList.add(item);
  }
/*
  public boolean removeItem(String name) {
    return groceryList.remove(new GroceryItem(name, 0, 0));
  }
*/
  public boolean removeItem(String name) {
    for (GroceryItem item : groceryList) {
      if(item.getName().equals(name)) {
        return groceryList.remove(item);
      }
    }
    return false;
  }

  public double getTotalCost() {
    double total = 0.0;
    for (GroceryItem item : groceryList) {
      total += item.getCost();
    }
    return total;
  }

  public String toString() {
    String retur = "";
    for (GroceryItem item : groceryList) {
```

```
            retur += item.toString() + "\n";
        }
        retur += "Samlet kostnad: " + getTotalCost();
        return retur;
    }
}
```

Alternativ 2:

d)
```
import java.util.ArrayList;

public class GroceryList extends ArrayList<GroceryItem> {

    public GroceryList() {
    }

    public boolean addItem(GroceryItem item) {
        return add(item);
    }

/*
    public boolean removeItem(String name) {
        return remove(new GroceryItem(name, 0, 0));
    }
*/
    public boolean removeItem(String name) {
        for (GroceryItem item : groceryList) {
            if(item.getName().equals(name)) {
                return remove(item);
            }
        }
        return false;
    }

    public double getTotalCost() {
        double total = 0.0;
        for (GroceryItem item : this) {
            total += item.getCost();
        }
        return total;
    }

    public String toString() {
        String retur = "";
        for (GroceryItem item : this) {
            retur += item.toString() + "\n";
        }
        retur += "Samlet kostnad: " + getTotalCost();
        return retur;
    }
}
```

Oppgave 2

```
import java.awt.BorderLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.*;

public class Valutakalkulator extends JFrame implements ActionListener {
  private JButton btnGbp, btnEur, btnUsd, btnSek, btnExit;
  private JTextField txtNok, txtResultat;
  private JLabel lblNok;
  private final double GBP = 8.873;
  private final double USD = 5.6;
  private final double EUR = 7.498;
  private final double SEK = 84.880;

  public Valutakalkulator() {
    setTitle("VALUTAKALKULATOR");
    JPanel pnlDisplay = new JPanel(new GridLayout(4, 2));
    lblNok = new JLabel("NOK");
    txtNok = new JTextField(12);
    txtResultat = new JTextField(12);
    btnGbp = new JButton("GBP");
    btnEur = new JButton("EUR");
    btnUsd = new JButton("USD");
    btnSek = new JButton("SEK");

    pnlDisplay.add(lblNok);
    pnlDisplay.add(txtNok);
    pnlDisplay.add(btnGbp);
    pnlDisplay.add(btnEur);
    pnlDisplay.add(btnUsd);
    pnlDisplay.add(btnSek);
    pnlDisplay.add(new JLabel("Resultat"));
    pnlDisplay.add(txtResultat);

    btnExit = new JButton("Avslutt");
    btnGbp.addActionListener(this);
    btnEur.addActionListener(this);
    btnUsd.addActionListener(this);
    btnSek.addActionListener(this);
    btnExit.addActionListener(this);

    add(pnlDisplay, BorderLayout.CENTER);
    add(btnExit, BorderLayout.SOUTH);

    setDefaultCloseOperation(EXIT_ON_CLOSE);
    setSize(400, 175);
    setVisible(true);
  }

  public static void main(String[] args) {
    new Valutakalkulator();
  }

  public void actionPerformed(ActionEvent e) {
    JButton kilde = (JButton) e.getSource();
    if(kilde == btnExit) {
      System.exit(0);
```

```
        }
        double nok = Double.parseDouble(txtNok.getText());
        double resultat = 0;
        double kurs = 0;
        String valuta = kilde.getActionCommand();
        String melding = "NOK " + nok + " =";

        if(kilde == btnGbp) {
          kurs = GBP;
        } else
        if (kilde == btnEur) {
          kurs = EUR;
        } else
        if (kilde == btnSek) {
          kurs = SEK / 100;
        } else {
          kurs = USD;
        }
        resultat = nok / kurs;
        String result = String.format("%5.2f", resultat);
        melding += " " + result + " " + valuta;
        txtResultat.setText(melding);
      }
    }
```

Oppgave 3

a)

```
Biggie a   JayZ a  Tupac b
Tupac b
Biggie

Tupac a
Tupac b
Tupac

JayZ a  Tupac b
Tupac b
Tupac

Biggie a   JayZ a  FiftyCent b
FiftyCent b
Biggie
```

b) NEI. Dette attributtet er deklarert `private` i klassen `Student`. Kan bare aksesseres via tilgangsmetoder

c) JA. Metoden er `public` i klassen `Student`. Kan kalles på direkte.

d)

```
public UndergraduateStudent(String name, int age, int year) {
  super(name, age);
  setYear(year);
}
```