

Vejledende løsninger til E00 eksamenssættet.

Opgave 1

Spørgsmål 1.1

FunnyStrings programmet giver følgende udskrift:

```
abc  
abcab  
012345
```

Spørgsmål 1.2

De to første opgaver klares nemmest med `String` klassens indbyggede metoder.

Spørgsmål 1.2.A

```
private static  
boolean isSenName (String name)  
{  
    return name.endsWith("sen");  
}
```

Spørgsmål 1.2.B

```
private static  
String change (String name)  
{  
    char[] tmp = name.toCharArray();  
    tmp[tmp.length-2] = 'o';  
  
    return new String(tmp);  
}
```

Spørgsmål 1.2.C

```
private static  
String toSwedish (String lastname)  
{  
    if (isSenName(lastname))  
        return change(lastname);  
    else
```

```
        return lastname;
    }
```

Opgave 2

Spørgsmål 2.1

Den løsning som ligger mest op ad kursets stil er:

```
private static
int minDivisor (int n)
{
    int i=2;
    boolean divisorFound = false;

    while (i<n && !divisorFound)
    {
        if (n%i == 0)
            divisorFound = true;
        else
            i=i+1;
    }
    return i;
}
```

Men det kan gøres noget kortere:

```
private static
int minDivisor (int n)
{
    int i=2;

    while (i<n && n%i!=0)
        i=i+1;
    return i;
}
```

Spørgsmål 2.2

Dette spørgsmål kan løses både iterativt og rekursivt. Den rekursive er ganske ligetil:

```
private static
String factorise(int n)
{
    int minDiv = minDivisor(n);
```

```
        if (minDiv==n)
            return Integer.toString(n);
        else
            return Integer.toString(minDiv) + "*" + factorise(n/minDiv);
    }
```

Den iterative løsning er lidt mere knudret:

```
private static
String factorise (int n)
{
    String res = "";

    if (n==1)
        res = Integer.toString(n);
    else
    {
        while (n>1)
        {
            int minDiv = minDivisor(n);

            if (minDiv==n)
                res = res + Integer.toString(n);
            else
                res = res + Integer.toString(minDiv) + "*";
            n = n/minDiv;
        }
    }
    return res;
}
```

Opgave 3

Spørgsmål 3.1

```
public
GUI_Exchange ()
{
    super("GUI_Exchange");
    // a panel for the text
    JPanel convertPanel = new JPanel();
    convertPanel.setLayout(
        new BoxLayout(convertPanel,BoxLayout.X_AXIS));

    euro          = new JTextField();
    JLabel text = new JLabel ("Euro equals $");
    dollars       = new JTextField();

    convertPanel.add(euro);
    convertPanel.add(text);
    convertPanel.add(dollars);

    // a panel for the buttons
    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(
        new BoxLayout(buttonPanel,BoxLayout.X_AXIS));

    euro2dollars = new JButton("Euro->$");
    clear        = new JButton("Clear");
    dollars2euro = new JButton("$->Euro");

    buttonPanel.add(euro2dollars);
    buttonPanel.add(clear);
    buttonPanel.add(dollars2euro);

    // the outer container
    Container contents = getContentPane();
    contents.setLayout(
        new BoxLayout(contents,BoxLayout.Y_AXIS));

    contents.add(convertPanel);
    contents.add(buttonPanel);

    // listener setup
    ActionListener listener = new MyListener();
    euro2dollars.addActionListener(listener);
    clear.addActionListener(listener);
    dollars2euro.addActionListener(listener);
}
```


Spørgsmål 3.2

MyListener klassen kunne laves på følgende måde:

```
private class MyListener implements ActionListener
{
    public
    void actionPerformed (ActionEvent event)
    {
        Object source = event.getSource();

        if (source == euro2dollars)
        {
            try
            {
                int value = Integer.parseInt(euro.getText());
                dollars.setText(
                    Integer.toString(Exchange.euro2dollars(value)));
            }
            catch (NumberFormatException exn) {};
        }

        if (source == dollars2euro)
        {
            try
            {
                int value = Integer.parseInt(dollars.getText());
                euro.setText(
                    Integer.toString(Exchange.dollars2euro(value)));
            }
            catch (NumberFormatException exn) {};
        }

        if (source == clear)
        {
            euro.setText("");
            dollars.setText("");
        }
    } // actionPerformed
}
```

Opgave 4

Spørgsmål 4.1

```
import Course;
import java.util.Vector;

public class Basic_Course implements Course
{
    private String name;
    private Course[] blocking;

    public
    Basic_Course(String name, Course[] blocking)
    {
        this.name=name;
        this.blocking=blocking;
    }

    public
    String name () { return name; }

    public
    Vector requirements () { return new Vector(); }

    public
    boolean isBlockedBy (Course course)
    {
        boolean res=false;

        if (blocking != null)
        {
            int i=0;

            while (i<blocking.length && !res)
            {
                res = course.name().equals(blocking[i].name());
                // with an equals method in the classes implementing
                // the course interface we can get away with:
                // res = course.equals(blocking[i]);
                i = i + 1;
            }
        }
        return res;
    }
} // Basic_Course
```

Spørgsmål 4.2

Den nemmeste løsning er nok at bruge en iterator:

```
boolean requirementsOK(Vector passed, Course c)
{
    boolean res = true;
    Iterator iter = c.requirements().iterator();

    while (res && iter.hasNext())
    {
        res = passed.contains(iter.next());
    }
    return res;
}
```

Man kan dog også behandle vektorerne mere som arrays:

```
boolean requirementsOK(Vector passed, Course c)
{
    boolean res = true;
    int index = 0;
    int size = c.requirements().size();

    while (res && index < size)
    {
        res = passed.contains(c.requirements().elementAt(index));
        index = index + 1;
    }
    return res;
}
```