

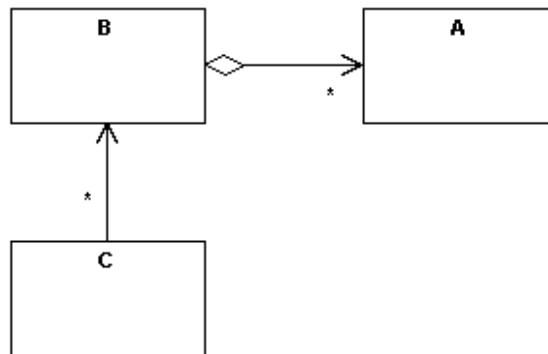
Lösningförslag till dugga

Kursnamn	Objektorienterade applikationer
Provdatum	2014-02-12
Program	DAI 2
Läsår	2013/2014, lp 3
Examinator	Uno Holmer

I lösningarna har den efterfrågade koden satts i fetstil.

Uppgift 1 (3+2 p)

a)



b)

```
public interface A;
public interface B extends A;
public abstract class C implements B;
public interface D;
public class E extends C implements D;
```

Uppgift 2 (5 p)

```
// Med anonym inre lyssnare
public class Gui extends JFrame {
    public Gui() { makeFrame(); }

    private void makeFrame() {
        final JTextField textField = new JTextField(20);
        textField.setEditable(false);
        add(textField);
         JButton button = new JButton("Update");
        button.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    textField.setText(computeSomething());
                }
            });
        add(button);
        pack();
        setVisible(true);
    }

    private String computeSomething () { ... }
}
```

```
// Med global lyssnare
public class Gui extends JFrame implements ActionListener {
    // OBS! Instansvariabel, annars ej synlig i lyssnarmetoden:
    private JTextField textField;

    public Gui() { makeFrame(); }

    private void makeFrame() {
        textField = new JTextField(20);
        textField.setEditable(false);
        add(textField);

        JButton button = new JButton("Update");
        button.addActionListener(this);
        add(button);

        pack();
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        textField.setText(computeSomething());
    }

    private String computeSomething () { ... }
}
```

Uppgift 3 (5 p)

```
public class Die extends Observable
{
    private int value;
    private static Random rand = new Random();

    public Die() { value = nextInt(); }
    public void roll() {
        value = nextInt();
        setChanged();
        notifyObservers(value);
    }
    public int getValue() { return value; }
    private int nextInt() { return rand.nextInt(6) + 1; }
}

public class Gui extends JFrame implements Observer
{
    private JTextField textField;
    private Die die;

    public Gui() {
        makeFrame();
        die = new Die();
        die.addObserver(this);
    }

    private void makeFrame() {
        JButton button = new JButton("Roll the die");
        button.addActionListener(
            new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                    die.roll();
                }
            });
        add(button);
        textField = new JTextField(3);
        textField.setEditable(false);
        add(textField);
        pack();
        setVisible(true);
    }

    public void update(Observable obj, Object arg) {
        if ( obj instanceof Die && arg instanceof Integer )
            textField.setText(""+(Integer)arg);
    }
}
```

Uppgift 4 (5 p)

```
public class Clock extends JFrame implements ActionListener {
    private JLabel clockLabel;
    private int seconds = 0;

    public Clock() {
        makeFrame();
    }

    private void makeFrame() {
        clockLabel = new JLabel();
        add(clockLabel);
        Timer timer = new Timer(1000,this);
        timer.start();
        pack();
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        clockLabel.setText(formatClockTime(seconds++));
    }

    private String formatClockTime(int seconds) {
        StringBuffer buf = new StringBuffer(8);
        buf.append(String.format("%2d:",seconds/3600));
        buf.append(String.format("%2d:",seconds/60%60));
        buf.append(String.format("%2d",seconds%60));
        return buf.toString();
    }
}
```

Uppgift 5 (5 p)

```
public class Main {
    public static void main(String[] arg) {
        List<Double> values = new ArrayList<Double>();
        for ( int i = 0; i < 10; i++ )
            values.add(new Double(i)); // Ex
        String fileName = arg[0];
        try {
            DataOutputStream out =
                new DataOutputStream(new FileOutputStream(fileName));
            out.writeInt(values.size());
            for ( Double x : values )
                out.writeDouble(x);
            out.close();
        }
        catch (IOException e) {
            ...
        }
    }
}
```

Uppgift 6 (5 p)

```
public class Main {

    public static String getRemote(String urlString)
    throws MalformedURLException, IOException
    {
        URL url = new URL(urlString);
        URLConnection uc = url.openConnection();
        BufferedReader in =
            new BufferedReader(
                new InputStreamReader(uc.getInputStream()));
        return in.readLine();
    }

}

// Alt.

public class Main {
    public static String getRemote(String urlString)
    throws MalformedURLException, IOException
    {
        URL url = new URL(urlString);
        URLConnection uc = url.openConnection();
        Scanner sc = new Scanner(uc.getInputStream());
        return sc.nextLine();
    }
}
```