

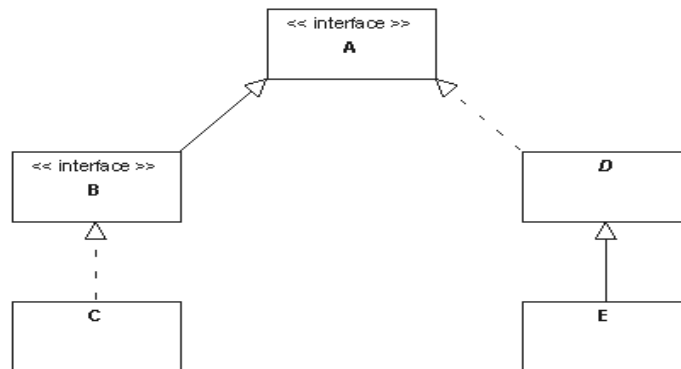
Lösningförslag till dugga

Kursnamn	Objektorienterade applikationer
Provdatum	2016-02-10
Program	DAI 2
Läsår	2015/2016, lp 3
Examinator	Uno Holmer

I lösningarna har den efterfrågade koden satts i **fetstil**.

Uppgift 1 (2+3 p)

a)



b) Relationerna i diagrammet ovan kan t.ex. realiseras så här i Java:

```
public class A {  
    private List<B> bs;  
    public A() { bs = new ArrayList<B>(); }  
    public void addB(B b) { bs.add(b); }  
}
```

```
public class B {  
    private C c;  
    public B(C c) { this.c = c; }  
}
```

```
public class C {  
    public void foo(E x) {}  
}
```

```
public class D {  
    private C c;  
    public D(C c) { this.c = c; }  
}
```

Uppgift 2 (5 p)

```
public class Account {
    private int balance;

    public Account(int balance) {
        this.balance = balance;
    }

    public synchronized
    void transaction(int amount) {
        while ( balance + amount < 0 ) {
            try {
                wait();
            }
            catch (InterruptedException e) {
                return;
            }
        }
        // assert: balance + amount >= 0
        balance += amount;
        notify();
    }
}
```

Uppgift 3 (5 p)

Alla detaljerna i layouthanteringen nedan krävs inte för full poäng.

```
public class Gui extends JFrame {
    private final int receivePort = 1234;
    private int timeout = 1000; // ms
    private JTextArea textArea;
    private DatagramReceiver datagramReceiver;

    public Gui(){
        makeFrame();
        datagramReceiver = new DatagramReceiver(receivePort);
    }

    private void receive() {
        String msg = datagramReceiver.receive(timeout);
        if ( msg == null )
            textArea.setText("No message received");
        else
            textArea.setText(msg);
    }

    private void makeFrame() {
        setTitle("Datagram receiver");
        makeMenuBar();
        textArea = new JTextArea(10,20);
        textArea.setEditable(false);
        add(textArea, BorderLayout.CENTER);
        JButton updateButton = new JButton("RECEIVE");
        updateButton.addActionListener(
            new ActionListener(){
                @Override
                public void actionPerformed(ActionEvent e) {
                    receive();
                }
            });
        JPanel p = new JPanel();
        p.setLayout(new FlowLayout());
        p.add(updateButton);
        add(p, BorderLayout.SOUTH);
        pack();
        setVisible(true);
    }
}
```

Uppgift 4 (5 p)

```
public class DatagramReceiver {
    private final int BUFSIZE = 1024;
    private DatagramSocket socket;
    private DatagramPacket packet;

    public DatagramReceiver(int port) {
        try {
            socket = new DatagramSocket(port);
            packet = new DatagramPacket(new byte[BUFSIZE],BUFSIZE);
        }
        catch (IOException e) {
            e.printStackTrace();
            System.exit(0);
        }
    }

    public String receive(int timeout) {
        try {
            socket.setSoTimeout(timeout);
            socket.receive(packet);
            return new String(packet.getData(),0,packet.getLength());
        }
        catch (Exception e) {
            return null;
        }
    }
}
```

Uppgift 5 (5 p)

```
public static Map<Integer,Float> readMaxTemperatures(String fileName)
{
    Map<Integer,Float> maxTemperatures = new TreeMap<Integer,Float>();
    try {
        DataInputStream in =
            new DataInputStream(new FileInputStream(fileName));
        long size = in.readLong();
        for ( long i = 0; i < size; i++ ) {
            int station = in.readInt();
            float temp = in.readFloat();
            if ( ! maxTemperatures.containsKey(station) ||
                maxTemperatures.get(station) < temp )
                maxTemperatures.put(station,temp);
        }
        in.close();
        return maxTemperatures;
    }
    catch (IOException e) {
        return null;
    }
}
```

Uppgift 6 (5 p)

```
public class Gui extends JFrame implements Observer
{
    private JTextField lineField,timeField;

    ...
    public void update(Observable obj,Object arg) {
        if ( obj instanceof Line && arg instanceof String )
            lineField.setText((String)arg);
        else if ( obj instanceof Time && arg instanceof String )
            timeField.setText((String)arg);
    }
}

public class Main {
    public static void main(String[] arg) {
        Time time = new Time();
        Line line = new Line();
        Gui gui = new Gui();
        line.addObserver(gui);
        time.addObserver(gui);
    }
}
```