

## Lösningsförslag till dugga

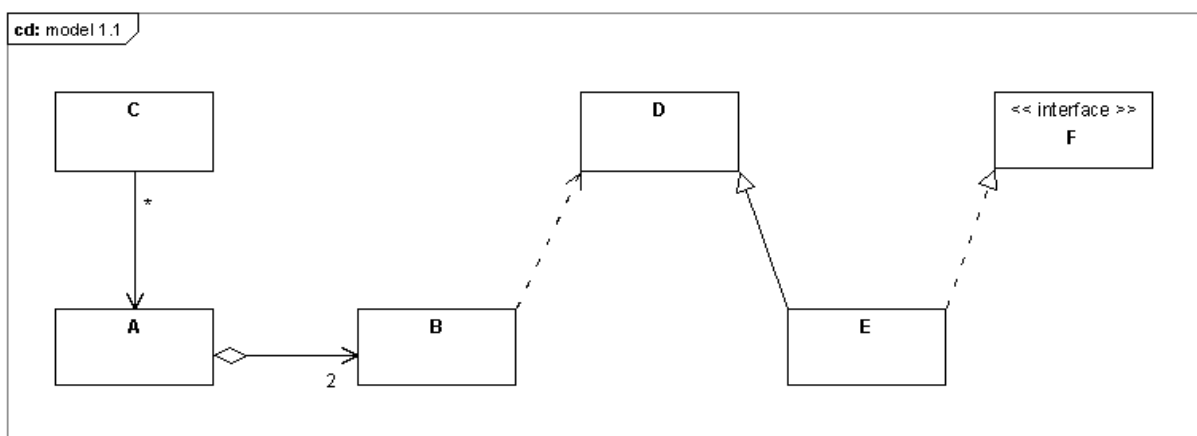
Kursnamn  
Provdatum

Objektorienterade applikationer  
2013-02-13

Program  
Läsår  
Examinator

DAI 2  
2012/2013, lp 3  
Uno Holmer

### Uppgift 1 (3 p)



### Uppgift 2 (4 p)

```
public MainWindow() {
    makeMenuBar();
    ...
}
...
private void makeMenuBar() {
    JMenuBar menuBar = new JMenuBar();
    setJMenuBar(menuBar);
    JMenu currencyMenu = new JMenu("Currency");
    menuBar.add(currencyMenu);
    ...
    JMenuItem euroItem = new JMenuItem("Euro");
    euroItem.addActionListener(
        new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                priceView.setText(format(price*SEK2EURO, "Euro"));
            }
        }
    );
    currencyMenu.add(euroItem);
}
```

**Uppgift 3** (3 p)

```
public void update(Observable o, Object arg) {
    if ( o instanceof A && arg instanceof String) {
        String s = (String)arg;
        System.out.println(s.toUpperCase());
    }
    if ( o instanceof B && arg instanceof Integer) {
        Integer i = (Integer)arg;
        System.out.println(i*10);
    }
}
```

**Uppgift 4** (1+3 p)

a)

Decorator.

b)

```
public class Singleton {
    private static Singleton instance = null;

    // Private prevents external object creation
    private Singleton() {}

    // Return the single object of this class
    // create (lazily) if necessary
    public static Singleton getInstance() {
        if ( instance == null )
            instance = new Singleton();

        return instance;
    }
    ...
}
```

**Uppgift 5** (2+3 p)

a)

```
DataInputStream in =
    new DataInputStream(new FileInputStream("infile.dat"));
```

```
PrintWriter out =
    new PrintWriter(new FileWriter("outfile.txt"));
```

b)

```
int size = in.readInt();
for ( int i = 0; i < size; i++ ) {
    long x = in.readLong();
    out.println(Long.toString(x));
}
```

**Uppgift 6** (4 p)

```
public class DoBackground extends Thread {
    private Computable obj;
    private float arg;
    private float value;
    private boolean ready = false;

    public DoBackground(Computable obj, float arg) {
        this.obj = obj;
        this.arg = arg;
        start();
    }

    public void run() {
        value = obj.compute(arg);
        ready = true;
    }

    public boolean isReady() {
        return ready;
    }

    public float getValue() throws IllegalStateException {
        if ( ! ready )
            throw new IllegalStateException();
        else
            return value;
    }
}
```

**Uppgift 7** (4 p)

```
public static void main(String[] arg) {
    String hostAddress = arg[0];
    int port = Integer.parseInt(arg[1]);
    try {
        InetAddress iadr = InetAddress.getByName(hostAddress);
        DatagramSocket socket = new DatagramSocket();
        Scanner in = new Scanner(System.in);
        while ( in.hasNextLine() ) {
            byte[] buf = in.nextLine().getBytes();
            socket.send(
                new DatagramPacket(buf, buf.length, iadr, port));
        }
    }
    catch (Exception e) {
        e.printStackTrace();
    }
}
```

**Uppgift 8** (3 p)

```
Class c = Class.forName("C");  
Method f = c.getMethod("f");  
Object o = c.newInstance();  
f.invoke(o);
```

alt.

```
( (Int)Class.forName("C").newInstance() ).f();
```