

OMD – Övning 3

Lösningar

Lösningarna presenteras ibland bara med implementeringar även i de fall där uppgiften efterfrågar diagram. På tentamen är det ej tillåtet att göra sådana avvikelser.

```
2 public class Switch extends Observable {
    private boolean on = false;

    public void toggle() {
        on = !on;
        setChanged();
        notifyObservers();
    }

    public boolean isOn() {
        return on;
    }
}

public class Control implements ActionListener {
    private Switch switch_;

    public Control(Switch switch_, View view) {
        this.switch_ = switch_;
        view.addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        switch_.toggle();
    }
}

public class View extends Button implements Observer {
    private Switch switch_;

    public View(Switch switch_) {
        super("OFF");
        this.switch_ = switch_;
        switch_.addObserver(this);
    }

    public void update(Observable o, Object arg) {
        setLabel(switch_.isOn() ? "ON" : "OFF");
    }
}
```

```

public class Frame extends JFrame {
    public Frame() {
        Switch switch_ = new Switch();
        View view = new View(switch_);
        Control control = new Control(switch_, view);
        add(view);
        setVisible(true);
        pack();
    }

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

```

3 Det cirkulära beroendet kan brytas genom att införa

```

public class WordOperand implements Operand {
    private hardware.Word word;
    // omissions
}

```

och modifiera Word:

```

public class Word {
    // omissions
}

```

Följande går också bra eftersom WordOperand inte utvidgar någon annan klass (föreslaget av en student i övningsgrupp E, 2012)

```

public class WordOperand extends hardware.Word implements Operand {
    // omissions
}

```

```

4 public interface WordFactory {
    public Word build(String string);
}

public class Factorial extends Program {
    public Factorial(WordFactory factory) {
        Address n = new Address(0),
            fac = new Address(1);
        add(new Copy(factory.build("5"), n));
        // omissions
    }
}

```