

Objektorienterad modellering och diskreta strukturer / design

Inför tentamen

Lennart Andersson

Reviderad 2012–10–18

2012

OMD 2012

F14-1

Computer projekttid

2011		2012	
timmar	antal	timmar	antal
2	1	2	0
4	4	4	0
6	8	6	4
8	4	8	13
10	10	10	9
12	6	12	11
14	4	14	2
16	6	16	12
18	1	18	2
20	0	20	10
21-	3	21-	4
snitt 11 tim		snitt 13,5 tim	

OMD 2012

F14-2

EDAF10 i HT2

Föreläsning måndag 29/10 kl 15.15-16.00 i MH:A.

Anmälan till XL-projektet via SAM.

Uppsamlingslaboration för EDAF10 måndagen den 29 oktober kl 8-10 i Alfa.

OMD 2012

F14-3

Rapporter från verkligheten

Jag har nyligen fått veta hur två olika företag tillämpar agila principer.

- ▶ Det ena rör ett företag i Silicon Valley. En av studenterna som följer den här kursen jobbade där under sommaren.
- ▶ Det andra är Google, som har ett stort kontor i London. En av doktoranderna på institutionen gjorde ett "internship" där under sommaren.

OMD 2012

F14-4

Silicon Valley

- ▶ Designprinciper tillämpas inte. Koden luktar illa.
- ▶ Huvuddelen av arbetet innebär modifiering av programkod och tillägg av kod för att möta nya kundkrav.
- ▶ Studenten tänker ta med sig kunskaperna från OMD-kursen till Silicon Valley nästa sommar.
- ▶ Jag tror inte att alla företag i området är lika oprofessionella. Jag tycker att studenten borde söka en bättre arbetsgivare.

OMD 2012

F14-5

Google

- ▶ Alla Googles program finns i 'molnet'.
- ▶ Det omfattar 100 miljoner rader kod.
- ▶ Ingen kod får läggas in i molnet förrän kollegor till programmeraren har granskat den och dålig design och illaluktande kod eliminerats. Jag tror att kontrollen är lika tuff som i OMD-projekten.
- ▶ Google försvarar sin plats som den mest attraktiva arbetsgivaren för data-civilingenjörer. Företaget erbjuder inte bara mycket goda arbetsförhållanden men har också en mycket professionell inställning till programvaruproduktion.

OMD 2012

F14-6


Designprinciper

- ▶ Lokalitetsprincipen
- ▶ Single Responsibility Principle
- ▶ Open/Closed Principle
- ▶ Dependency Inversion Principle (Abstraktionsprincipen)
- ▶ Integritetsprincipen
- ▶ Liskov Substitution Principle
- ▶ Interface Segregation Principle

OMD 2012

F14-9

Designmönster

- ▶ Command [Computer]
- ▶ Composite [XL, lab 2]
- ▶ Template Method [Computer, lab 2]
- ▶ Strategy [nästan överallt, Computer, XL]
- ▶ Decorator [Övning 2]
- ▶ Global Object (aka, also known as Singleton) 
- ▶ Null Object [XL?]
- ▶ Observer [XL]
- ▶ Factory Method [XL, lab 4]
- ▶ Interpreter [Computer]

OMD 2012

F14-10

Litteratur

På tentamen får medföras

Martin, PPP

Andersson, Föreläsningsbilder F01-06.pdf

Andersson, UML

Java snabbreferens

Andersson, Diskreta strukturer [EDAF10]

Sedvanliga anteckningar är OK. Information kopierad från andra dokument får inte förekomma.

OMD 2012

F14-11

Tentamenslokal

Tentamen är torsdagen den 25 oktober kl 8.00 i Vic: 1A-D.

OMD 2012

F14-12

Om du vill lära dig mer

- ▶ EDA270 Programvaruutveckling i grupp — projekt, fokus på utvecklingsprocessen XP med rika tillfällen att göra design.
- ▶ EDA180 Kompilator teknik, mycket modellering, aspektorienterad programmering, programmets mest avancerade programmeringsprojekt.
- ▶ EDAN40 Funktionsprogrammering, en annan paradig.
- ▶ EDAN01 Constraint programming, ytterligare en paradig.

OMD 2012

F14-13

Uppgift 1

Följande två klasser är väldigt lika. Använd *Template method*-mönstret för att eliminera duplicerad kod och det gemensamma attributet. Lösningen redovisas med ett klassdiagram som visar arv och associationer och med programkod.

OMD 2012

F14-14

Uppgift 1

```
public class StartButton extends Button
    implements ActionListener {
    private Train train;
    public StartButton(Train train) {
        super("Start");
        this.train = train;
        addActionListener(this);
    }
    public void actionPerformed(ActionEvent event) {
        train.start();
    }
}
```

OMD 2012

F14-15

Uppgift 1

```
public class StoppButton extends Button
    implements ActionListener {
    private Train train;
    public StoppButton(Train train) {
        super("Stopp");
        this.train = train;
        addActionListener(this);
    }
    public void actionPerformed(ActionEvent event) {
        train.stopp();
    }
}
```

OMD 2012

F14-16

Uppgift 2

I en modell av en liten dator representeras hoppinstruktionen med (se nästa bild) där `address` är den absoluta adressen till den instruktion man skall hoppa till. Man vill nu modifiera modellen så att man kan använda både absoluta och relativa adresser. En relativ adress innebär att man skall hoppa bakåt eller framåt ett givet antal instruktioner. I en textmässig representation skriver man absoluta adresser utan tecken och relativa med tecken:

```
JMP    38
JMP    +3
JMP    -3
```

OMD 2012

F14-17

Uppgift 2

```
public class Jump implements Instruction {
    private int address;

    public Jump(int address) {
        this.address = address;
    }

    public void execute(Memory memory, ProgramCounter prog) {
        programCounter.setCounter(address);
    }

    public String toString() {
        return "JMP" + ' ' + address;
    }
}
```

OMD 2012

F14-18

Uppgift 2

Gör en design som respekterar *open/closed*-principen så att man lätt kan lägga till denna och andra adresseringsformer utan att ändra i gammal kod eller definiera nya hoppinstruktioner.

Lösningen visas med Java-kod för den nya Jump-klassen och det som behövs för att kunna representera absoluta och relativa adresser.

OMD 2012

F14-19

Uppgift 3

Följande klasser ingår i ett röstningsprogram med ett grafiskt användargränssnitt. Modifiera programmet så att *Observer*-mönstret används för att uppdatera vyn så att modellen inte känner till den konkreta vyn. Lösningen redovisas med Java-kod.

OMD 2012

F14-20

Uppgift 3

```
public class Counters {
    private int yesCounter, noCounter;
    private View view;
    public Counters(View view) {
        this.view = view;
    }
    public void incrementYes() {
        yesCounter++;
        updateView();
    }
    public void incrementNo() {
        noCounter++;
        updateView();
    }
    private void updateView() {
        view.update(yesCounter, noCounter);
    }
}
```

OMD 2012

F14-21

Uppgift 3

```
public class View extends JPanel {
    public void update(int yesCounter, int noCounter) {
        // omissions
    }
}
```

OMD 2012

F14-22

Uppgift 3

I java.util finns

```
public interface Observer {
    public void update(Observable observable,
                      Object object);
}

public class Observable {
    public void addObserver(Observer observer) ...
    protected void setChanged() ...
    public void notifyObservers(Object object) ...
}
```

OMD 2012

F14-23

Uppgift 4

I klassen `ArrayList` returnerar metoden `toString` en sträng som innehåller strängrepresentationen av alla elementen separerade av ", ". Strängen inleds med "[" och avslutas med "]".

Utvidga klassen så att man kan utnyttja *Strategy*-mönstret för att välja vilken inledande, separerande och avslutande sträng som skall användas.

Lösningen redovisas som Java-kod för den utvidgade klassen, strategi-gränssnittet och en strategi-klass som ger samma sträng som `toString` i `ArrayList`.

Man kan traversera listan t ex genom att använda den `Iterator` som metoden `iterator` i `ArrayList` returnerar. En `iterator` har metoderna `hasNext()` och `next()`.

OMD 2012

F14-24

Diskreta strukturer

Förstå, konstruera och manipulera

- ▶ satslogiska uttryck
- ▶ predikatlogiska uttryck
- ▶ mängduttryck
- ▶ språk
- ▶ reguljära uttryck
- ▶ grammatiker
- ▶ relationer
- ▶ funktioner

OMD 2012

F14-25

Diskreta strukturer

Kunna utföra

- ▶ naturlig härledning
- ▶ substitution
- ▶ härledning med grammatik
- ▶ syntaxanalys

OMD 2012

F14-26

Tentamen 12-08-24: 4a

Det reguljära uttrycket $0^*(10^+)^*(1 | \epsilon)$ beskriver ett språk på alfabetet $\{0, 1\}$. Vilka av följande strängar tillhör språket?

01100 01010 ϵ 1001100

OMD 2012

F14-27

Tentamen 12-08-24: 4b

Ange alla strängar med längden 4 som tillhör det språk som genereras av grammatiken

$S ::= SS$
 $S ::= aSa | bSb | \epsilon$

OMD 2012

F14-28

Tentamen 12-08-24: 5a

Visa att $\{p \rightarrow q, \neg q\} \vdash \neg p$ genom att komplettera följande härledning.

$$\frac{p \rightarrow q}{\quad} \quad \frac{\quad}{\quad} \neg q$$

OMD 2012

F14-29

Tentamen 12-08-24: 5b

Vi definierar två predikat på mängden av naturliga tal

$p(x, y) \triangleq x$ är delbart med y

$q(x, y) \triangleq x = y^2$

Konstruera med hjälp av dessa ett predikatlogiskt uttryck som säger att ett godtyckligt tal är delbart med 4 om det är jämt och kvadraten på ett tal.

OMD 2012

F14-30

Tentamen 11-10-21: 5b

$\rho \triangleq \{(0, 1), (1, 1), (1, 2)\}$ är en relation på $\{0, 1, 2\}$. Beräkna ρ^0 , ρ^2 och ρ^* .

Tentamen 09-10-22: 5a

Låt $F(x, y)$ betyda att x är far till y , $M(x, y)$ att x är mor till y och $E(x, y)$ att x och y är samma person. Konstruera ett predikat som är sant precis då x och y är syskon. Två personer är syskon om de har samma föräldrar men är olika personer.

Tentamen 08-10-23: 5b

Ge ett exempel på när $e[x \setminus e_1][y \setminus e_2]$, där e , e_1 och e_2 är aritmetiska uttryck, inte är lika med $e[y \setminus e_2][x \setminus e_1]$.