

Tentamen i EDAF25

1 juni 2017

Skrivtid: 14-19

- SKRIV BARA PÅ ENA SIDAN AV PAPPRET – tentorna kommer att scannas in, och endast framsidorna rättas.
- SKRIV **INTE** MED FÄRGPENNA – enda tillåtna färg är svart/blyerts.
- SKRIV TYDLIGT – om texten inte går att läsa kan du inte få några poäng.
- SÄTT IDENTITET OCH SIDNUMMER PÅ VARJE INLÄMNAT BLAD, kontrollera att sidnumret på din sista sida är samma som det antal blad du markerar på omslagspappret.

Hjälpmedel:

- Koffman & Wolfgang: *Data Structures*
- Martin: *Agile Software Development*
- Andersson: UML-syntax
- Holm: Java snabbreferens

Uppgift 1

En espressomaskin har två vippströmbrytare: A (huvudströmbrytare och värmeelement) och B (pumpen). Två utsignaler ska styras: värmeelement (V-on och V-off) och pump (P-on och P-off). Pumpen får inte slås på om inte värmeelementet är på. Rita ett fullständigt UML-tillståndsdigram för espressomaskinen.

2 p

Uppgift 2

Nedan, i figur 1, visas ett exempel på ett designmönster.

- Vilket designmönster visar exemplet?
- Rita ett objektdiagram där ett objekt av varje shapetyp ingår.
- Rita ett sekvensdiagram för scenariet att metoden draw i Drawing anropas för ditt objektdiagram.

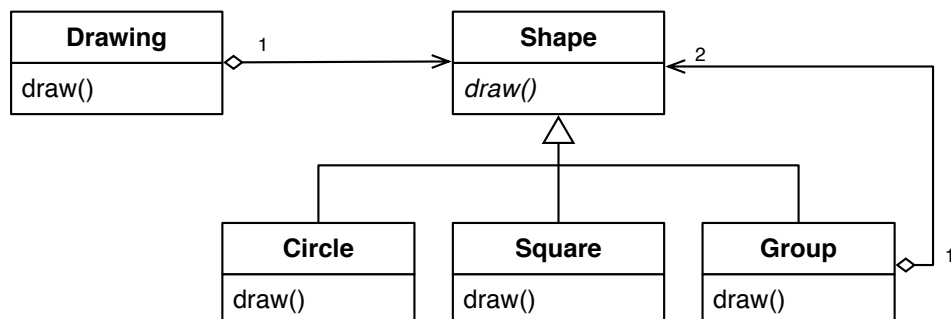


Figure 1: Klassdiagram Drawing

3 p

Uppgift 3

Använd *Strategy*-mönstret för att konstruera en klass `FilteredList`, som innehåller en lista av strängar och ett "filter" som gör att metoden `toString` bara returnerar en sträng sammansatt av de strängar som filtret accepterar.

Om listan innehåller ["good", "morning", "sir"] och man använder ett filter som accepterar strängar med minst fyra tecken skall `toString` returnera

```
good
morning
```

Om filtret accepterar alla strängar skall resultatet bli

```
good
morning
sir
```

Filtret skall vara ett attribut i klassen `FilteredList`. Lösningen redovisas som ett Java-program som innehåller allt som behövs för att exekvera exemplet med två utskrifter.

4 p

Uppgift 4

I figur 2 nedan visas ett försök att designa ett ramverk (framework) med en användaranpassning i form av två egna specialiseringar av `Shape`. Designen på ramverket har dock en brist vilken kan avhjälpas genom att applicera mönstret Fabriksmetod (Factory method).

- (a) Vad är det för brist?
- (b) Applicera Fabriksmetod-mönstret genom att komplettera (och ändra om det behövs) klassdiagrammet.
- (c) Implementera (i Java) metoden "create"!

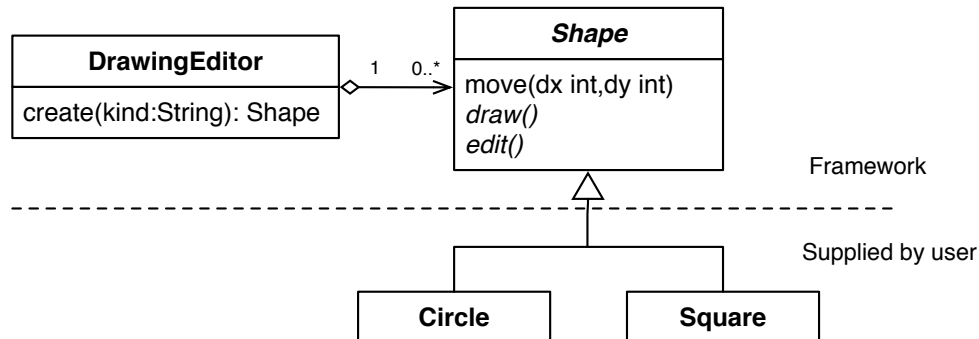


Figure 2: Ramverk DrawingEditor

4 p

Uppgift 5

Följande klass agerar både modell, vy och kontroll (control).

```
public class Switch extends JButton implements ActionListener {
    private boolean on = false;

    public Switch() {
        super("OFF");
        addActionListener(this);
    }

    public void actionPerformed(ActionEvent e) {
        on = !on;
        setLabel(on ? "ON" : "OFF");
    }
}
```

Gör om designen så att MVC-mönstret används med tre separata klasser med användning av *Observer*-ramverket.

- (a) Lösningen redovisas med Java-kod och skall visa hur och var följande metoder används.

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

public abstract class Observable {
    public void addObserver(Observer observer)
    protected void setChanged()
    public void notifyObservers()
}
```

- (b) Rita ett fullständigt klassdiagram av lösningen där dina tre klasser finns med, samt klassen Observable och gränssnittet Observer.

6 p

Uppgift 6

Grafen i figur 3 visar kapaciteten i ledningarna i ett flödesnätverk.

4 p

- (a) Räkna ut det maximala flödet i nätverket.
(b) Beskriv och namnge algoritmen du använder.
(c) Visa också hur nätverket kan representeras med hjälp av en matris.

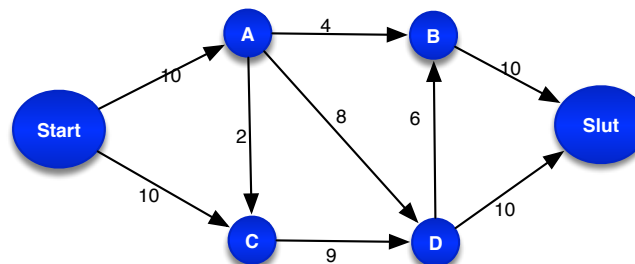


Figure 3: Flödesnätverk

Uppgift 7

Denna uppgift innehåller ett antal deluppgifter, där varje deluppgift har ett påstående och en anledning. För varje deluppgift, svara med ett av följande alternativ:

- A Både påståendet och anledningen är korrekta uttalanden och anledningen förklarar påståendet på ett korrekt sätt.
- B Både påståendet och anledningen är korrekta uttalanden, men anledningen förklarar inte påståendet.
- C Påståendet är ett korrekt uttalande, men anledningen är ett felaktigt uttalande.
- D Påståendet är felaktigt, men anledningen är ett korrekt uttalande.
- E Både påståendet och anledningen är felaktiga uttalanden.

Svara inte i uppgiftshäftet, utan skriv ditt svar på ett vanligt lösningspapper. Varje korrekt besvarad deluppgift ger 1.0 poäng – vi vill inte att ni skall chansa, så ni får -0.25 poäng på en deluppgift om ni svarar fel, men aldrig minuspoäng totalt sett på hela uppgift 1.

	Påstående	Anledning
T1	Javakoden i figur 4 visar ett exempel på mallmetod-mönstret (Template method).	Klassen Slot i figur 4 är en abstrakt klass med två mallmetoder.
T2	Javakoden i figur 4 visar ett exempel på dekoratörmönstret (Decorator).	Subklasserna CommentSlot och ExpressionSlot kan lägga till funktionalitet till en instans av Slot utan att andra instanser av Slot påverkas.
T3	Det går att bryta mot SRP utan bryta mot DRY och vice versa.	En klass kan ha flera ansvar utan att duplicera kod och två klasser kan lösa olika problem på samma sätt.
T4	Om du har kod som bryter mot LSP bör du överväga om du kan använda delegation, komposition eller aggregering som ett alternativ till generalisering.	LSP säger att generalisering bör undvikas.
T5	Det är svårt att göra ändringar i en stel (rigid) design.	En stel design innebär att det finns många beroenden till en klass.
T6	Genom att använda kompositmönstret bryter man mot SRP.	Kompositmönstret låter användaren behandla hela samlingar och enskilda objekt på ett enhetligt sätt.
T7	För att tillämpa SAP (Stable-Abstractions)-principen på model-paketet i figur 4 bör man lägga den abstrakta klassen Slot i ett annat paket.	Genom att flytta Slot till ett annat paket blir model-paketet mer instabilt

7p

/ Nedan beskrivs 3 klasser som ingår i ett paket, model. Det finns ytterligare 7 klasser i detta paket. Paketet är i sin tur ett av 5 paket i ett program som hanterar och visualiserar kalkylark "XL". Paketet model har i denna kontext en abstrakthet som är 0.1 och en instabilitet som är 0.2. Ett 30-tal klasser i andra paket har beroenden till model-paketet. */*

```
abstract class Slot {
    private String slot;
    private String text;
    public abstract double getValue(Environment env);
    public abstract String toString(Environment env);
}

class ExprSlot extends Slot {
    ExprSlot(String slot, Expr expr){
        super(slot, expr.toString());
        this.expr = expr;
    }
    @Override
    double getValue(Environment env) {
        return expr.value(env);
    }
    @Override
    String toString(Environment env) {
        NumberAdjustment adj = new NumberAdjustment(0,1);
        return adj.left(getValue(env)+"");
    }
}

public class CommentSlot extends Slot {
    CommentSlot(String slot, String comment){
        super(slot, comment);
    }
    @Override
    double getValue(Environment env) {
        return 0;
    }
    @Override
    String toString(Environment env) {
        String s = getText();
        return s.substring(1);
    }
}
```

Figure 4: Slot