

Tentamen i Objektorienterad modellering och diskreta strukturer

Tentamen består av 6 uppgifter med totalt 30 poäng. För godkänt betyg kommer att krävas högst 16 poäng. Vid bedömningen kommer hänsyn att tas till lösningens kvalitet. Onödigt komplicerade lösningar kan ge poängavdrag. UML-diagram skall ritas i enlighet med UML-häftet.

Hjälpmedel:

Martin: Agile Software Development

Andersson: Diskreta strukturer

Andersson: UML-syntax

Föreläsningbilderna F01-13.pdf

Holm: Java snabbreferens

Java snabbreferens och UML-syntax finns att låna hos skrivningsvakten.

1 I en modell av en liten dator finns följande huvudklass.

```
public class Computer {
    private Memory memory;
    private Program program;
    private ProgramCounter programCounter;
    public void run() {
        programCounter.setCounter(0);
        while (programCounter.getCounter() >= 0) {
            Instruction instruction = program.get(programCounter.getCounter());
            programCounter.increment();
            instruction.execute(memory, programCounter);
        }
    }
}
```

Använd *Strategy*-mönstret för att ge möjlighet att logga de instruktioner som exekveras på olika sätt. Modifiera klassen `Computer` och lägg till klasser/gränssnitt så att man kan bestämma vilken strategi som skall användas innan man anropar den parameterlösa metoden `run()`. Lösningen ges som Java-kod och skall innehålla två strategier: en som sparar exekverade instruktioner i en lista och en som inte gör någonting.

2 I samma modell finns ett minne som innehåller heltal med 64 bitar och klassen `Add` som representerar en additionsinstruktion.

```
public class Memory {
    private long memory[];
    public long getValue(int address) {
        return memory[address];
    }
    public void set(int address, long value) {
        memory[address] = value;
    }
}
```

```

    }
    public int size() {
        return memory.length;
    }
}

public class Add implements Instruction {
    private int address1, address2, address3;
    public void execute(Memory memory, ProgramCounter programCounter) {
        long value1 = memory.getValue(address1);
        long value2 = memory.getValue(address2);
        memory.set(address3, value1 + value2);
    }
}

```

Gör om designen så att man har ett minne där varje adress svarar mot ett heltal givet av 128 bitar. Den nya designen skall vara anpassad för liknande förändringar. Lösningen redovisas med ett klassdiagram med modifierade och tillfogade klasser och gränssnitt. Diagrammet skall innehålla alla attribut och metoder som behövs för att man skall kunna implementera `execute` i Add-klassen. Metoderna skall uppvisa typerna för parametrar och eventuellt returvärde..

3 I Add-klassen i uppgift 2 finns också

```

    public String toString() {
        StringBuilder builder = new StringBuilder();
        builder.append("ADD ");
        builder.append(address1);
        builder.append(" ");
        builder.append(address2);
        builder.append(" ");
        builder.append(address3);
        return builder.toString();
    }

```

Det finns en analog Mul-instruktionsklass med operationskoden MUL. Använd *Template Method*-mönstret för att eliminera duplicerade attribut och duplicerad kod. Lösningen redovisas med Java-kod.

4 Mängden av reguljära uttryck på alfabetet Σ definieras av

- \emptyset är ett reguljärt uttryck
- om $\sigma \in \Sigma$ så är σ ett reguljärt uttryck
- om α och β är reguljära uttryck så är $(\alpha \cdot \beta)$ ett reguljärt uttryck
- om α och β är reguljära uttryck så är $(\alpha \mid \beta)$ ett reguljärt uttryck
- om α är ett reguljärt uttryck så är α^* ett reguljärt uttryck

Gör en objektorienterad modell för reguljära uttryck. Man får förutsätta att alla symboler i Σ går att representera med datatypen `char` i Java. Lösningen redovisas med Java-kod. Inga konstruerare eller metoder behöver implementeras.

- 5 a. Visa $\{p \rightarrow q\} \vdash \neg q \rightarrow \neg p$ genom att komplettera följande bevissträd. Indikera i varje nod vilken inferensregel som används.

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

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

- b. Låt $even(n)$ vara sant precis då n är ett jämt tal. Använd predikatlogik för påståendet att det finns minst två jämna tal i \mathbb{N} .
- 6 a. Gör en BNF-grammatik för de reguljära uttrycken i uppgift 4.
- b. Låt ρ vara en relation på $\mathbb{N}_3 \triangleq \{0, 1, 2\}$ definierad som $\rho \triangleq \{(0, 1), (1, 2), (2, 0)\}$. Beräkna ρ^3 .