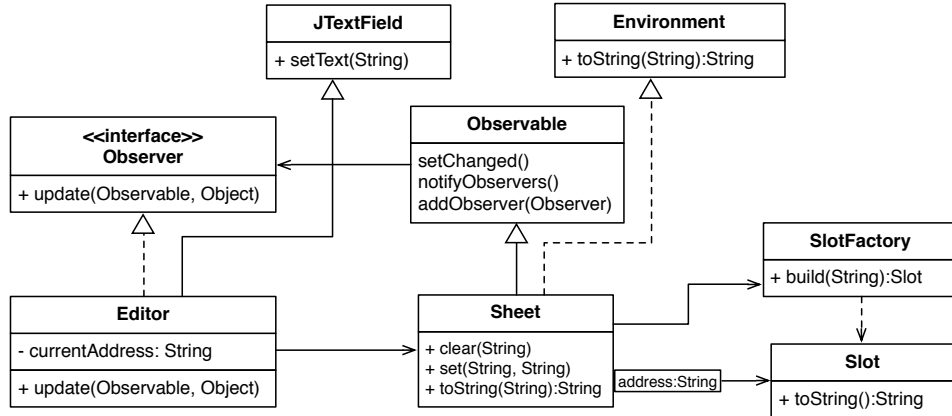


Tentamen i Objektorienterad modellering och design

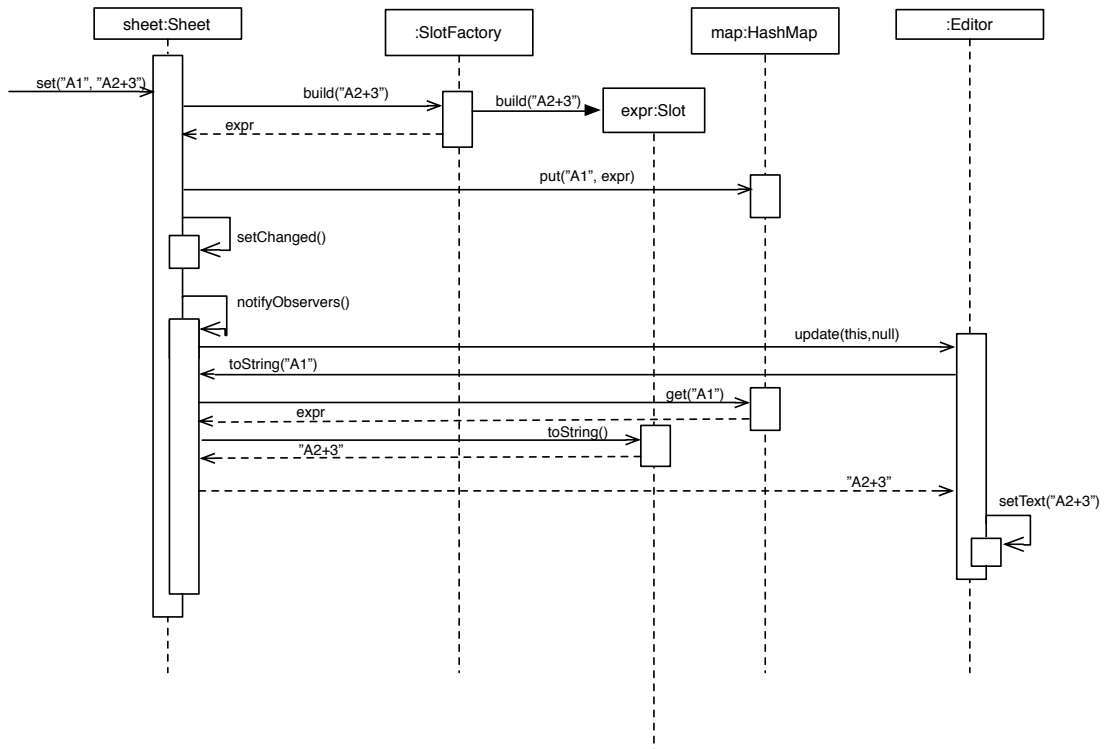
Lösningar

1. a. Klassdiagram

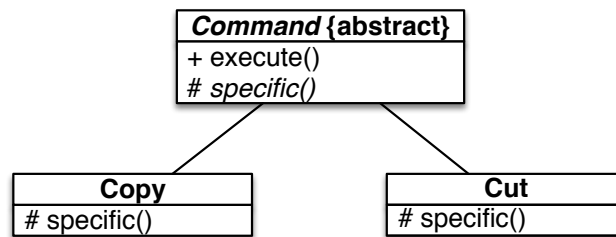


b. Sekvensdiagram

Visar ett exempel där currentAddress i Editorn är "A1".

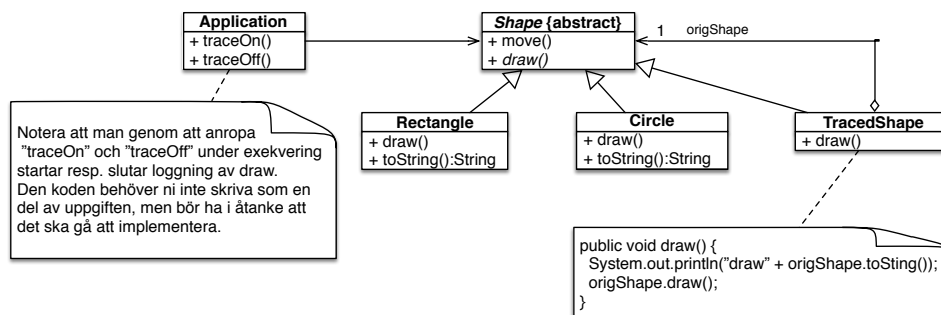


2. a. Klassdiagram



b. Metoden execute() i Command ändras till konkret och i den läggs koden från Copy och Cut. Den del av koden som skiljer sig åt ersätts av ett anrop till den nya metoden specific. specific är abstrakt i Command. Den specifika koden läggs i den konkreta implementation av specific i Copy resp. Cut. specific ska ej vara publikt åtkomlig utan görs protected.

3. a. Lösningen följer inte Open/Closed-principen. Om man vill lägga till fler Shapes, t ex en Square, så behöver man även implementera en TracedSquare. Svårt implementera traceOn() och traceOff(), då man måste skapa nya objekt varje gång, kopiera över aktuella tillstånd i existerande figurer.



b.

```

4. public class Account {
    private Revenue revenue;
    private float balance;
    public void setRevenue(Revenue revenue) {
        this.revenue = revenue;
    }
    public float revenue(int days) {
        return revenue.amount(days, balance);
    }
}
  
```

```

public interface Revenue {
    public float amount(int days, float balance);
}
public class Standard implements Revenue {
    public float amount(int days, float balance) {
        return balance*days*4.0/365;
    }
}
  
```

5. Tillståndsdigram

