

# Laboration 4

Casper Beckman

December 31, 2025

## Contents

### 1 Uppgift 1

Vi har fått i uppgift att implementera Composite-mönstret. Vad detta är kan ses i uppgiftslydelsen. Vi skriver först ett interface för **Component**, vilken kommer bli implementerad av både Composite och Component.

```
<Component.java>≡  
    String toString();  
    double getWeight();  
    String getName();
```

Här är **getName** endast relevant för extrauppgiften. Notera att vi ej har deklarerat funktionerna som endast skulle användas i **Component**. Vi implementerar dessa i **AbstractComponent.java**

```
<AbstractComponent.java>≡  
    import java.util.ArrayList;  
    public class AbstractComponent implements Component {  
        protected double weight;  
        protected String name;  
        protected ArrayList <Component> children; // this is always empty for leaf nodes  
  
        public double getWeight() {  
            <AbstractComponent getWeight>  
        }  
  
        public String toString() {  
            <AbstractComponent toString>  
        }  
        public String getName() {  
            return name;  
        }  
    }
```

Vi noterar att vi vill att alla komponenter som innehålls i en komponent ska vara med i `getWeight` respektive `toString`, därmed är det naturligt att implementera dessa rekursivt.

```
<AbstractComponent getWeight>≡  
    double sum = weight;  
    if (children != null) {  
        for (Component c : children) {  
            sum+=c.getWeight();  
        }  
    }  
    return sum;
```

Notera att denna implementation ger en "trailing newline", dock påverkar detta inte oss i vår implementation därmed bryr vi oss inte om att göra en mer korrekt implementation.

```
<AbstractComponent toString>≡  
    StringBuilder sb = new StringBuilder();  
    sb.append(name).append("\n");  
    if (children != null) {  
        for (Component c : children) {  
            sb.append(c.toString());  
        }  
    }  
    return sb.toString();
```