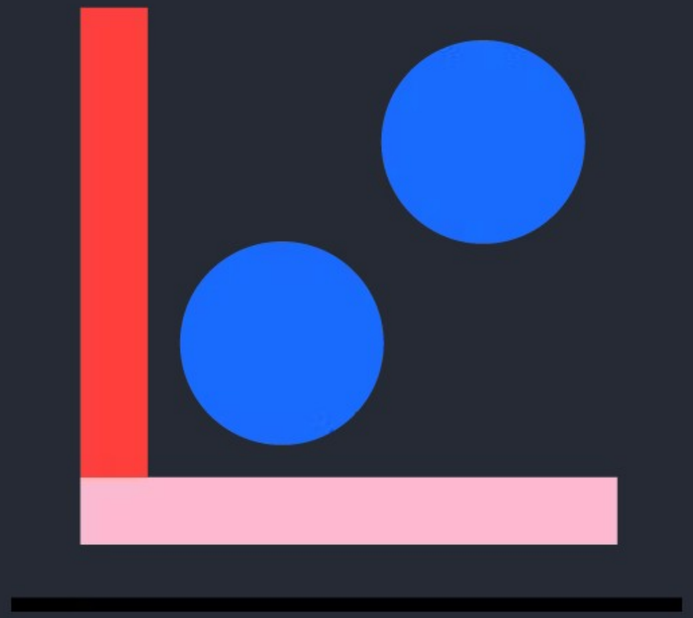
Mentimeter

Designmønstre

IN1010 gruppetime 9

Gruppelærer: Sigmund Trutt



Mentimeter

Instructions



Mentimeter

Agenda

- Iterator
- Observer
- Decorator
- Singleton

Mentimeter

Designmønstre

Mentimeter

Iterator-mønsteret

- Har to innebygde metoder:
  - hasNext() og next()
- Brukes for å gå gjennom forskjellige typer beholdere
  - f. eks array og ArrayList

```java
ArrayList<Integer> tallListe = new ArrayList<>();
tallListe.add(e:5);
tallListe.add(e:1);
tallListe.add(e:10);
tallListe.add(e:16);
tallListe.add(-5);
tallListe.add(e:18);
tallListe.add(e:2);

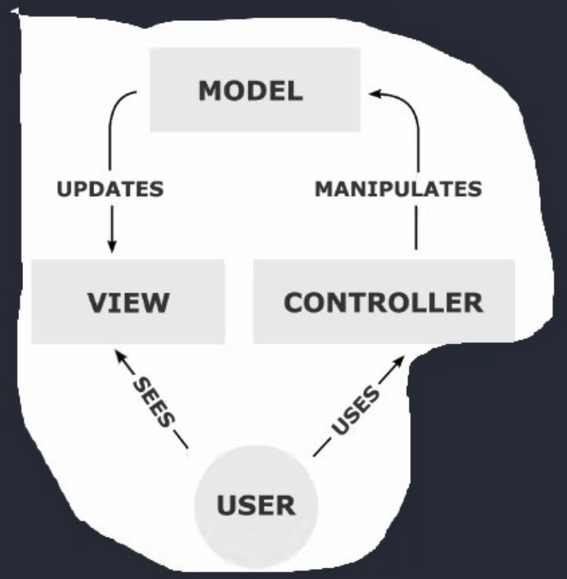
//ALTERNATIV 1 - Lage et Iterator-objekt
Iterator<Integer> iterator = tallListe.iterator();
while (iterator.hasNext()) {
    System.out.println(iterator.next());
}


//ALTERNATIV 2 - Bruke forEach metoden:
for (int tall : tallListe) {
    System.out.println(tall);
}
```

Mentimeter

Observer-mønsteret

- Et ***observator***-objekt observerer ***subjekt***-objekter og får beskjed om endringer
- Eksempel:
  - Model-View-Controller (MVC)
- Nyhetsfeed: NRK legger ut en ny artikkel -> Observer oppdaterer nettsiden -> følgere blir notifisert



Mentimeter

Decorator-mønsteret

- Legger til/fjerner egenskaper til et objekt
- Eksempler: Bestilling av mat på nett - Du bygger opp en bestilling med tillegg
- Kan lage decorator for hver ekstra-ting, og så legge på disse som man vil
- Eksempel med PizzaLager

Mentimeter

Singleton-mønsteret

- Et Singleton -> et *unikt* objekt av en gitt type i programmet
  - Kan kun lage ett objekt av denne typen
- Eksempel på bruk:
  - Utskrift til terminal:
  - Hadde flere objekter skrevet ut til terminal samtidig risikerer vi at objekter som kjører samtidig skriver ut om hverandre

Mentimeter

Singleton-eksempel:

Singleton brukt til terminal-logging

```java
public class SingletonLogger {

    // Kun én instans, dermed kan vi bruke static
    private static SingletonLogger instance = null;

    //Privat konstruktør slik at ingen andre klasser kan lage et objekt av den
    private SingletonLogger() {    }

    // Metode for å lage nøyaktig én instans dersom det ikke eksisterer fra før
    public static SingletonLogger getInstance() {
        if (instance == null) instance = new SingletonLogger();
        return instance;
    }


    // Metode som logger til terminalen
    public void logMessage(String message) {
        System.out.println("Log: "+ message);
    }

}
```

Mentimeter

Egenarbeid

- Innlevering 3 - rekursjon
- Ukesoppgaver

Mentimeter

Ask me anything

0 questions
0 upvotes