

Föreläsning 3

Objektorienterad programmering DD1332

Accessor, designmönster

1

public, private, protected och paketprivat

- **public:** åtkomlig för alla
- **private:** åtkomlig bara i klassen
- **protected:** åtkomlig i modulen och subklasser
- **ingetdera (paketprivat):** åtkomlig i modulen

2

Design av klass

- **Inkapsling:** dölja attributen i en klass och ha kontroll över åtkomst till de via metoder.
- **Lös koppling:** Klasserna är väl inkapslade, har minimal antal referenser till varandra och begränsar användning av API.
- **Hög sammanhållning:** Att klassens attribut och metoder stödjer så att klassen får en tydlig roll .

3

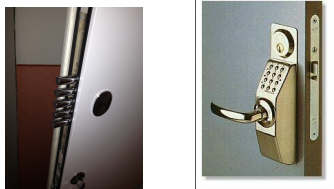
Exempel

```
public class TV{
    private int aktuellKanal;
    ...
    public void setKanal(int nyKanal){
        if (nyKanal>0 || nyKanal<100)
            aktuellKanal=nyKanal;
    }
}
```

4

Designmönster

- Exempel: kodlås eller säkerhetsdörr



5

Designmönster

- Exempel: Skriv en klass som ska föreställa månen i vårt solsystem så att när den används i ett solsystemprogram, ska det inte gå att skapa flera instanser av månen (single tone)

6

Designmönster

```
public class Moon{
    private static Moon instance = null;

    public Moon(){
        Moon.getInstance();
    }

    private static Moon getInstance() {
        if (instance == null)
            instance = new Moon();
        return instance;
    }
}
```

7

Designmönster

Designmönster är alltså ett generellt tankesätt som kan användas för att kunna hitta en lösning till ett förekommande problem av samma typ.

8

Designmönster

Designmönster är alltså ett generellt tankesätt som kan användas för att kunna hitta en lösning till ett förekommande problem av samma typ.

9

Polymorfi

- En operator som kan operera på fler typer kallas polymorf till skillnad mot monomorfa operatorer som endast kan operera på en typ.

10

Överladdning av metoder

```
class Calculator{
    int addTerm(int x, int y ){
        return x+y;
    }
    double addTerm(double x, double y){
        return x+y;
    }
}
```

11

Arv

- Möjligheten att koden i en klass kan användas i en annan klass.
- Viktigt att det är arvning ska vara meningsfull
- Det görs m.h.a. extends
- En klass kan ära från en enda klass.

12

Is-A och Has-A

- Is-A: Används vid deklaration. Ett sätt att säga den här klassen är en typ av den där klassen. När en klass ärver från en annan. T.ex: Volvo Is-A Car
- HAS-A: Det förekommer när man vill använda det man har redan deklarerat. Cruise ship Has-A lifeboat

13

Överskrivning av metoder

När man deklarerar en metod med exakt samma paramterar i subklassen då har man gjort en överskrivning av metoden.

14