

# EDA016 Programmeringsteknik för D

## Läsvecka 4: Systemutveckling

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

## 4 Systemutveckling

- Att göra denna vecka
- Oföränderlighet (immutability)

# Att göra i Vecka 4: Förstå hur systemutveckling går till med klasser och objekt i en integrerad utvecklingsmiljö

- 1 Läs följande kapitel i kursboken:  
2.7–2.10, 3.3–3.12  
Begrepp: oföränderlighet
- 2 Gör övning 4: ???
- 3 Träffas i samlarbetsgrupper och hjälp varandra förstå
- 4 Gör Lab 3: ???

# Standardvärden för attribut

# Oföränderlighet (immutability)

# Förhindra att variabler **ändras** med **final**

Attributet `latinsktNamn` nedan är en **konstant**.

Kompilatorn hjälper oss att kolla så att vi inte råkar ändra på det vi har deklarerat som **final**.

```
1  class Gurka {
2      public int vikt = 100; //gram
3
4      public final String latinsktNamn = "Cucumis sativus";    // *1
5
6      public String visa() {
7          System.out.println("Denna gurka (" + latinsktNamn + ") väger " + vikt + "g");
8      }
9  }
10
11  public class Constant {
12      public static void main(String[] args){
13          Gurka g = new Gurka();
14          g.vikt = 200;
15          g.latinsktNamn = "Tomat";    // ERROR: ger kompileringsfel! Vilket?
16          g.visa();
17      }
18  }
19
20  // *1: final deklarereras gärna även static om det bara behövs en enda
```

# Oföränderligt objekt

```
1  class Gurka { // exempel på oföränderligt objekt (eng. immutable objekt)
2      private int vikt;
3
4      void Gurka(int vikt) {
5          this.vikt = vikt; //endast här tilldelas attributet ett värde
6      }
7
8      public Gurka halva(){ // förändrar inte denna instans, skapa ny istället
9          return new Gurka(vikt/2);
10     }
11
12     public void visa() {
13         System.out.println("Denna gurk-instans väger för alltid " + vikt + " gram");
14     }
15 }
16
17 public class ImmutableObject {
18     public static void main(String[] args){
19         Gurka g1 = new Gurka(42);
20         Gurka g2 = g1; // g1 och g2 refererar till samma objekt
21         g1 = g1.halva(); // förändringen av g1 påverkar inte g2
22         g1.visa();
23         g2.visa();
24     }
25 }
```