

# EDA016 Programmeringsteknik för D

## Läsvecka 15: Avslutning, Utblick

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

## 15 Avslutning, Utblick

- Att göra denna vecka
- Om tentamen
- Inlämningsuppgift
- Kursutvärdering
- Grumligtlådan
- Utblick
- Frågestund
- Om nästa läsperiod: Gäster Anna Axelsson & Martin Höst
- Avslutning och tack för mig!

## Att göra i Vecka 15: Repetition, uppsamling, tentaplugg.

- 1 Uppsamling: **gör klart och redovisa kvarvarande labbar och inlämningsuppg. på veckans resurstider**
- 2 Träffas i samarbetsgruppen & hjälp varandra att tentaplugga.
- 3 Ingen labbtid denna vecka - använd resurstider för uppsamling och frågor inför tentapluggande.
- 4 Kolla på tentapluggtipsen på **förra föreläsningen**.

# Extenta: Föreläsning på genomgång av Sociala-nätverkstentan

- Tenta: [150318.pdf](#)
- Lösning: [sol-150318.pdf](#)

Frågor på uppgift 1-3?

# Extenta: Försättning på genomgång av Sociala-nätverkstentan

- Tenta: [150318.pdf](#)
- Lösning: [sol-150318.pdf](#)

## Frågor på uppgift 1-3?

**Uppgift 4:** Antag att det sociala nätverket i föregående uppgift ska vidareutvecklas för att stödja en ny typ av användare. Man vill nu att både företag och privatpersoner ska kunna använda nätverket och man vill lägga till detta utan att ändra i någon av klasserna i tidigare uppgifter. Företag ska, förutom namn, id, aktiviteter och vänlista (som en "vanlig" User har) även ha organisationsnummer och slogan. Vi gör förändringen genom att införa en ny klass.

- Beskriv hur förändringen bäst kan införas i det projekt som beskrivs i tentan. En korrekt förklaring ger 2 poäng.
- Skriv kod för att visa hur den nya klassen ser ut. Lösningen behöver inte innehålla några andra metoder förutom konstruktorn. Korrekt kod ger ytterligare 3 poäng.

# Tenta: anmälning och anonyma tentor

- Du **måste vara godkänd** på **alla** labbar + inlämningsuppg. för att få tenta.
- Läs **alla** instruktioner **noga** och **anmäl dig** här:  
<http://www.student.lth.se/studieinformation/anonyma-tentor/>
- **Gruppbonus**: Din grupps samarbetsbonus meddelas via mejl. Kontakta mig om något verkar felaktigt.

## Inlämningsuppgift Bank; diskussion + utblick

### Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

## Inlämningsuppgift Bank; diskussion + utblick

### Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

Genom att implementera logik som förhindrar för stora uttag.



# Inlämningsuppgift Bank; diskussion + utblick

## Exempel på klurighet

Specifikationen tillåter uttag som är större än saldot på kontot, men i det textuella användargränssnittet ska felmeddelande ges vid för stora uttag som inte medges.

Hur löser vi det?

Genom att implementera logik som förhindrar för stora uttag.

- När man använder ett api behöver man förstå vad api:et omfattar och hur det är tänkt att användas.
  - Vad behöver jag själv kolla?
  - Vad kollas av koden i api:et som jag använder?
  - Vilken klass ansvarar för vad?
- Mer om design-frågor i OMD-kursen.

# Inlämningsuppgift Draw; diskussion + utblick

## Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `se.lth.cs.pt.shapes.Shape` har inga metoder som exponerar attributen `x` och `y`.

Hur löser vi det?

# Inlämningsuppgift Draw; diskussion + utblick

## Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `se.lth.cs.pt.shapes.Shape` har inga metoder som exponerar attributen `x` och `y`.

Hur löser vi det? Två olika lösningar:

- Kopiera `se.lth.cs.pt.shapes.Shapes` till mitt eget paket och lägg till metoderna `getX()`, `getY()` eller `deltaMove(dx, dy)`
- Eftersom attributen `x` och `y` är **protected** i `Shape` så kan vi med en ny klass `LocatableShape` **extends** `Shape` och implementera `getX()`, `getY()` eller `deltaMove(dx, dy)` i denna

# Inlämningsuppgift Draw; diskussion + utblick

## Exempel på klurighet

Utvidgning: relativ förflyttning med tangentbordet.

Men klassen `se.lth.cs.pt.shapes.Shape` har inga metoder som exponerar attributen `x` och `y`.

Hur löser vi det? Två olika lösningar:

- Kopiera `se.lth.cs.pt.shapes.Shapes` till mitt eget paket och lägg till metoderna `getX()`, `getY()` eller `deltaMove(dx, dy)`
- Eftersom attributen `x` och `y` är **protected** i `Shape` så kan vi med en ny klass `LocatableShape` **extends** `Shape` och implementera `getX()`, `getY()` eller `deltaMove(dx, dy)` i denna

Viktiga generella observationer:

- Om du har tillgång till källkoden och får kopiera den kan du bygga vidare på koden (en s.k. **fork**), men vad göra om api:et kommer i ny version?
- Genom arv kanske det går att fixa det du saknar, men vad göra om api:et kommer i ny version?

# CEQ – Course Experience Questionnaire

- Görs på hela LTH på samma sätt. Alla får länkar via mejl.
- Snälla fyll i CEQ! Jag är **mycket tacksam** för all konstruktiv feedback! Hög svarsfrekvens är viktigt för att kunna dra slutsatser om variationen i svaren och signifikansen i sammanställningen.
- Del 1: Generella påståenden, alla med 5-gradig skala: tar helt avstånd ... instämmer helt
- Del 2: **Fritextfrågor**:  
"Vad tycker du var det bästa med den här kursen?"  
"Vad tycker du främst behöver förbättras?"
- Fördel med CEQ: Samma alla kurser alla år medger jämförelse över tid.
- Begränsning med CEQ: Saknar frågor kopplat till specifika kursmoment.
- Mer om CEQ här: <https://www.ceq.lth.se/>

## Kursspecifik utvärdering om specifika kursmoment

- Jag vill gärna att alla gör den LTH-gemensamma, anonyma kursutvärderingsenkäten **CEQ**. Dina fritext-kommentarer om vad som är det bästa med kursen och vad som främst behöver förbättra emottages mycket tacksamt i CEQ-utvärderingen!
- Jag vill också komplettera CEQ med en kursspecifik utvärdering av specifika kursmoment i denna kurs och jag är därför **mycket tacksam** om alla fyller i denna snarast: <http://goo.gl/forms/x5ClcMX22Q>
- Jag behandlar dina svar konfidentiellt, men ber om din STiL-id så att jag kan återkomma om jag mot förmodan undrar något mer.
- Din input är mycket värdefull vid framtida kursutveckling!

# Några avslutande lappar ur Grumligtlådan

- **overloading**: flera metoder med samma namn
- **override**: ersätta metod i subklass
- **iterator**: ett objekt med metoder för att iterera över elementen i en samling
- "När du säger **Java** exakt vad menar du då?"
  - 1 Ett programspråk
  - 2 En virtuell maskin och exekveringsplattform
  - 3 En stor uppsättning standardbibliotek med färdiga klasser
  - 4 En global gemenskap kring ett stort antal öppen-källkodsbibliotek
- **volatile**: speciellt nyckelord för skydd mot samtidig access av gemensamma variabler i jämlöpande trådar.

Java har speciella språkkonstruktioner och bibliotek för att hantera trådar som exekverar samtidigt. Detta är ett avancerat och intressant område som ni läser mer om i realtidsprogrammering.

# Utblick



# Utblick: Vart är Java och JVM-teknologin på väg?

- En viktig trend är kombinationen av objekt-orientering och funktionsprogrammering och detta visar sig i **Java 8** med sina anonyma funktioner (s.k. lambda expressions) och nya samlingen `java.util.stream`.
- I **Java 9** som resultat av "**Project Jigsaw**" kommer mycket handla om modularisering av JDK som möjliggör skräddarsydda, mindre runtime-versioner av JVM.
- Mycket av utmaningarna handlar om att dra nytta av många kärnor och parallellprogrammering blir allt hetare. Därför jobbar man mer och mer med oföränderlig data som gör det lättare att dela upp beräkningar på flera kärnor.
- Det finns flera livaktiga språk som kör på JVM, t.ex. **Scala** och **Clojure** men också **Javascript** med skriptmotorn **Nashorn**
- Med kompilatorn för Scala till Javascript **Scala.js** kan man dela kod mellan en server som kör på JVM och en klient som kör i webbläsaren.

# Utblick: Vart är software engineering på väg?

Några faktorer som skapar förändringstryck för ökad produktivitet i utvecklingsprocessen för mjukvara:

- Snabbast vinner.
- Jakt på kompetens.
- Sakernas internet.

Några starka trender:

- Open source
- Big Data
- Devops

# Frågestund

# Om nästa läsperiod: Gäster Anna Axelsson & Martin Höst

# Hoppas att kursen varit kul och lärorik!



# Ett stort TACK för...

- ... att ni kämpat så glatt!
- ... att ni ställt massor med frågor!
- ... att det har varit så hög närvaro på föreläsningarna!
- ... att ni är så konstruktiva och verkligen vill lära er!

## Ett stort TACK för...

- ... att ni kämpat så glatt!
- ... att ni ställt massor med frågor!
- ... att det har varit så hög närvaro på föreläsningarna!
- ... att ni är så konstruktiva och verkligen vill lära er!

**Ett stort LYCKA TILL på vägen till att bli en kompetent och innovativ systemutvecklare!**