EDA016 Programmeringsteknik för D

Läsvecka 1: Introduktion

Björn Regnell

Datavetenskap, LTH

Lp1-2, HT 2015

1 Introduktion

- Om denna kurs
- Vad är programmering?
- Vårt första Java-program
- Grundläggande programkonstruktioner i Java
- Sammanfattning
- Meddelande från Code@LTH

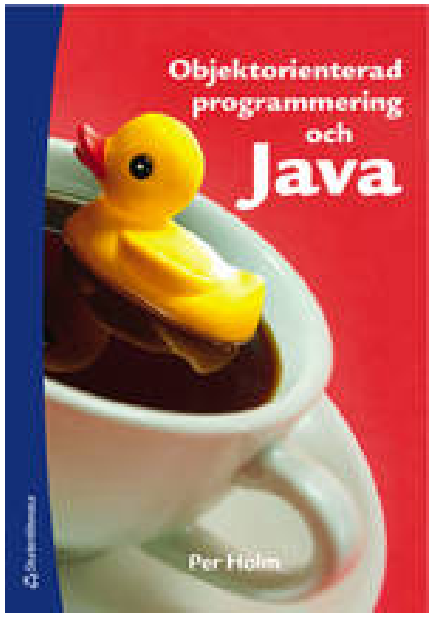
Om denna kurs

Vad och hur?

- *Vad* ska du lära dig?
  - Grundläggande principer för programmering
  - Konstruktion av enkla algoritmer
  - Tänka i abstraktioner
  - Imperativ och objektorienterad programmering
  - Programspråket Java
  - Utvecklingsmiljön Eclipse: implementera, testa, felsöka
- *Hur* ska du lära dig?
  - Genom praktiskt eget arbete: Lära genom att göra!
  - Genom studier av kursens teori: Skapa förståelse
  - Genom samarbete med dina kurskamrater

Kurslitteratur

- ”Objektorienterad programmering och Java" av Per Holm
- Kurskompendium med övningar och laborationer
- Bokpaket säljs på KFS
  John Ericssons väg 4
  `http://www.kfsab.se/`



Personal

Kursansvarig:

Björn Regnell, bjorn.regnell@cs.lth.se

Kurssekreterare:

Lena Ohlsson
Exp.tid 09.30 – 11.30 samt 12.45 – 13.30

Handledare:

Maj Stenmark, Tekn. Lic., Doktorand
Gustav Cedersjö, Doktorand
Anton Klarén, D09
Maria Priisalu , D11
Anders Buhl, D13
Erik Bjäreholt, D13
Fatima Abou Alpha, D13
Cecilia Lindskog, D14
Emma Asklund, D14

Kursmoment — varför?

- **Föreläsningar**: skapa översikt, ge struktur, förklara teori, svara på frågor, motivera varför
- **Övningar**: bearbeta teorin med avgränsade problem som mestadels löses med papper & penna, förberedelse inför laborationerna
- **Laborationer**: lösa programmeringsproblem praktiskt, obligatoriska uppgifter; lösningar redovisas för handledare
- **Resurstider**: få hjälp med övningar och laborationsförberedelser av handledare
- **Samarbetsgrupper**: grupplärande genom samarbete och dialog
- **Kontrollskrivning**: obligatorisk, diagnostisk, kamraträttad; kan ge samarbetsbonuspoäng till tentan
- **Inlämningsupgift**: du visar att du kan skapa ett större program självständigt; redovisas för handledare
- **Tenta**: Skriftlig tentamen utan hjälpmedel, förutom snabbreferens.

Nytt för i år

Årets kurs är i flera avseende väsentligt annorlunda and förra årets upplaga, så lita inte på allt som era äldre kursare säger :)

- Övningar blir resurstider i datorsal
- Inlämningsuppgift utan skriftlig rapport
- Samarbetskultur och grupplärande
- Nya övningar
- Nya laborationer
- Nya föreläsningar

Ändringarna är framtagna i samråd med studierådet. Mer om bakgrunden här:

http://fileadmin.cs.lth.se/cs/Education/EDA016/2015/update.pdf

Detta är bara början...

Exempel på efterföljande kurser som bygger vidare på denna:

- Årskurs 1
  - Programmeringsteknik – fördjupningskurs
  - Utvärdering av programvarusystem
  - Diskreta strukturer
- Årskurs 2
  - Objektorienterad modellering och design
  - Programvaruutveckling i grupp
  - Algoritmer, datastrukturer och komplexitet
  - Funktionsprogrammering

Förkunskapsenkät

http://goo.gl/forms/agwUg2uHqM

Samarbetsgrupper

En typisk kursvecka

- Gå på föreläsningar måndag–tisdag
- Jobba själv med boken, övningar, labbförberedelser
- Träffas i samarbetsgruppen och hjälp varandra att förstå mer och fördjupa lärandet
- Gå till resurstider och få hjälp och tips av handledare, onsdag–torsdag
- Genomför obligatorisk laboration på fredagen

Se detaljerna och undantagen i schemat i TimeEdit

Resurstider

Laborationer

hej

Att skapa koden som styr världen

I stort sett alla delar av samhället styrs av mjukvara:

- kommunikation
- transport
- byggsektorn
- statsförvaltning
- finanssektorn
- media
- sjukvård
- övervakning
- integritet
- upphovsrätt
- miljö & energi
- sociala relationer
- utbildning

Hur blir ditt framtida yrkesliv som systemutvecklare?

- Redan nu är det en skriande brist på utvecklare och bristen blir bara värre och värre...
  CS 2015-08-17
- Global kompetensmarknad
  CS 2015-06-14
  CS 2015-08-15
- Fokus på innovation, tid-till-marknad
- Autonoma utvecklingsteam i decentraliserade organisationer
- Öppen källkod

Att göra i Vecka 1: Rivstarta

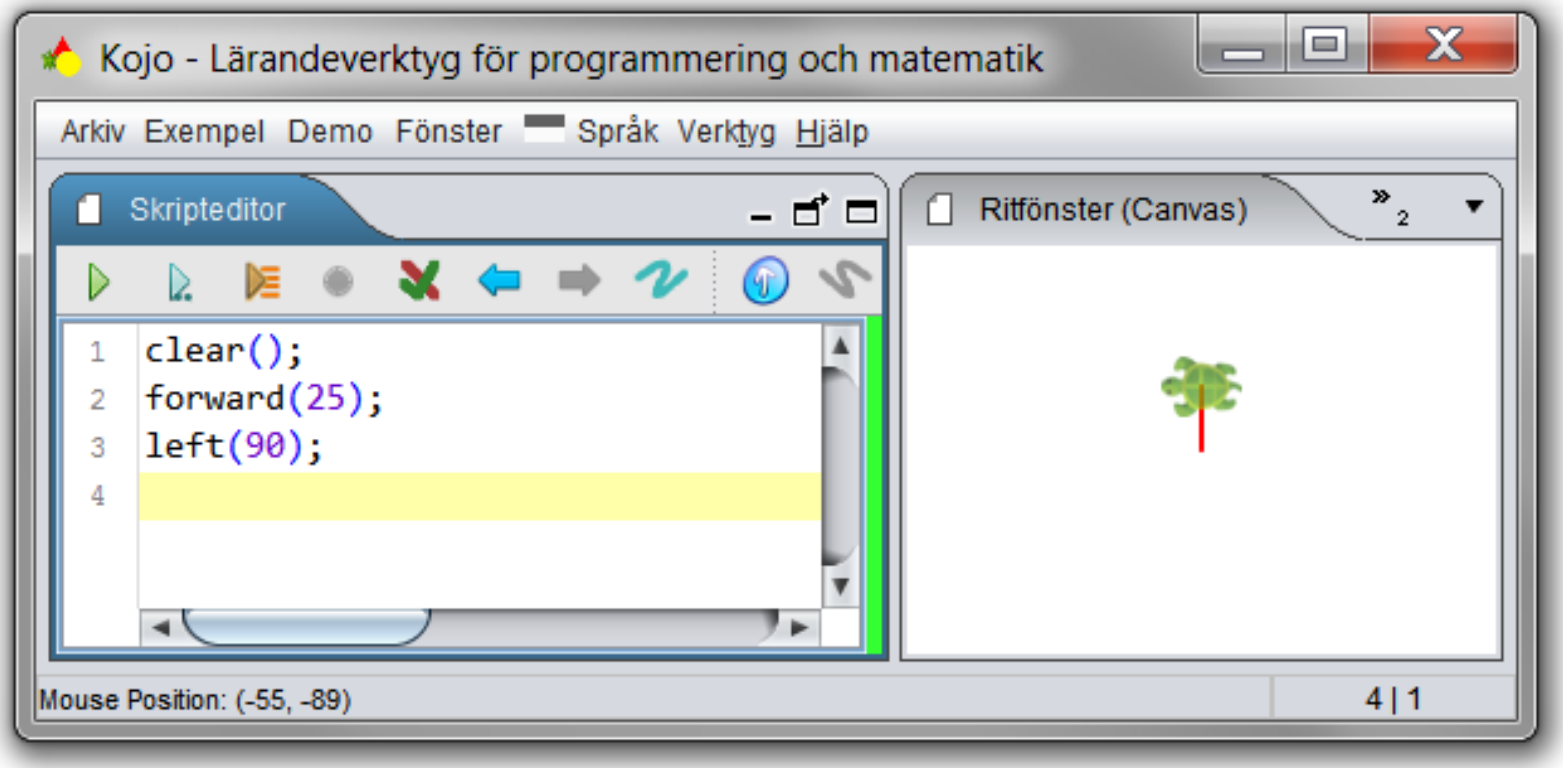
1. Läs följande kapitel i kursboken:
   1, 3.1-3.3, 5.1-5.3, 6.1-6.2, 7.1-7.3, 7.5-7.6, 7.8-7.9
2. Gör övning 1: Hello World, Hello Args, javac, editera–kompilera–exekvera–felsök, värden, uttryck variabler och tilldelning
3. Gör förberedelserna till laboration 1: skapa textfil, etc.
4. Träffas i samarbetsgrupper och hjälp varandra att förstå
5. Kom till **resurstiderna** och få hjälp och tips
6. Genomför laboration 1: Quiz — träna på att editera, läsa, ändra och felsöka i färdig programkod

   `while`, `for`, `Scanner`

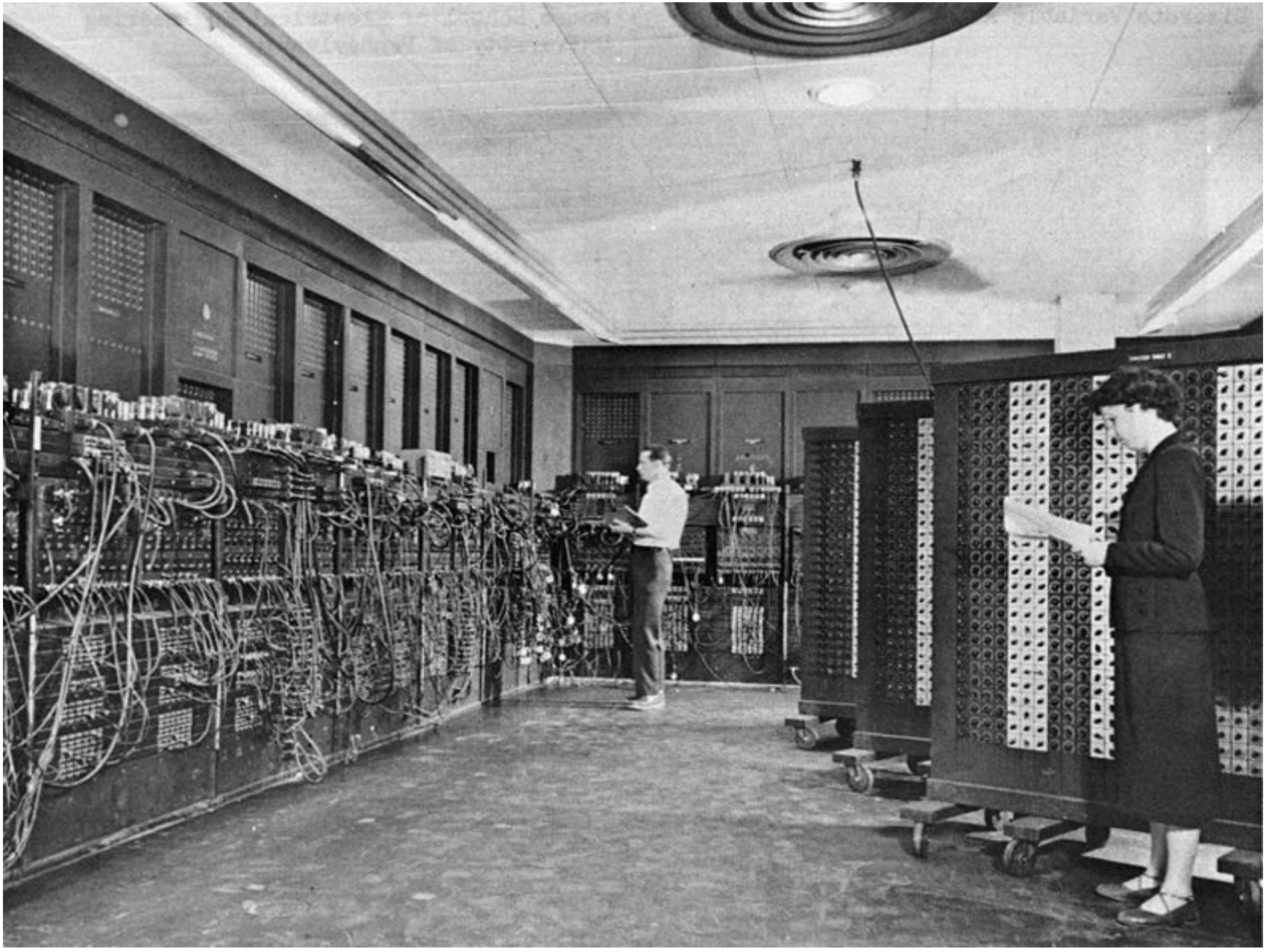
Vad är programmering?

Programming unplugged

Editera och exekvera ett program i Kojo



Vad är en dator?



Vad är en kompilator och ett programspråk?

Vad är Java?

Utvecklingsverktyg

Vad är objektorientering?

- Det finns många olika programmeringsparadigm (sätt att programmera på), till exempel:
  - **imperativ programmering:** programmet är uppbyggt av sekvenser av olika satser som påverkar systemets tillstånd
  - **objektorienterad programmering:** en sorts imperativ programmering där programmet består av objekt som sammanför data och operationer på dessa data
  - **funktionsprogrammering:** programmet är uppbyggt av samverkande (matematiska) funktioner som undviker föränderlig data och tillståndsändringar
  - **logikprogrammering:** programmet är uppbyggt av logiska uttryck som beskriver olika fakta eller villkor och exekveringen utgörs av en bevisprocedur som söker efter värden som uppfyller fakta och villkor

Grundläggande principer i imperativ programmering

- **Sekvens**: Ett program innehåller sekvenser av *satser*. Ordningen mellan dessa har helt avgörande betydelse.
- **Alternativ**: Systemet reagerar på vad som händer och kan välja olika vägar genom programmet beroende på *variablers* värde
  Java: `if`-sats eller `switch`-sats
- **Repetition**: Göra saker om och om igen
  Java: `while`-loop eller `for`-loop
- **Abstraktion**: Kapsla in (komplexa) programdelar och sätta namn på dessa så att de enkelt går att återanvända utan att att vi behöver ”rota i inanndömet”.
  Java: klasser och metoder

Hello World!

Vårt första Java-program i filen `HelloWorld.java`

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hej och välkomna!");
    }
}
```

Kompilera och kör:

```
> javac HelloWorld.java
> java HelloWorld
Hej och välkomna!
>
```

Ovan ingår i övning 1.

Hello World! – Vad betyder egentligen allt detta?

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hej och välkomna!");
    }
}
```

- **`public`** Denna programdel är synlig ”utåt” och kan användas av andra delar.
- **`class`** Ett slags "kodbyggblock" som samlar olika programdelar. All java-kod måste finnas i en klass. Det finns tusentals färdiga klasser att använda direkt i Java och man kan lätt skapa egna klasser. Klammerpar { } anger början och slut.
- **`static`** Denna programdel skapas direkt vid programmets start och det finns exakt *en* sådan här per klass.
- **`void`** Berättar för kompilatorn att inget värde returneras från denna programdel.
- `main` Berättar var exekveringen av programmet börjar.
- `( )` Parentespar berättar för kompilatorn att vi här kan ha parametrar.
- `String[] args` Möjliggör indata till programmet i form av flera textsträngar. Parametern `args` måste finnas i `main`, men vi använder den inte i detta program.
- `System.out.println` Den färdiga klassen `System` kan bl.a. skriva ut text. Textsträngar avgränsas av citationstecken. Semikolon avgränsar satser.

Översikt av några grundläggande programkonstruktioner i Java

- värde (value): data som programmet kan använda
  `42` `"hej"` `42.0` `true`
- uttryck (expression): data kombineras med operatorer och ger nya värden
  `41+1` `"h"+"ej"` `43.5-1.5` `!false`
- deklaration av variabel (variable declaration): skapa plats i minnet för data
  `int x = 42;`
- tilldelningssats (assignment): ändra värdet på variabler
  `x = 43;`
- alternativ (choice): välj väg beroende på variablers värde
  `if switch`
- repetition (loop): upprepa om och om igen
  `while for`

Värden och uttryck

Meddelande från Code@LTH