

Objektorienterad Programmering (TDDC77)

Föreläsning I: kursinfo, att programmera datorer, första programmet

Ahmed Rezine

IDA, Linköpings Universitet

Hösttermin 2017



Kursinfo

- ▶ hemsida: <http://www.ida.liu.se/~TDDC77>
- ▶ imperativ och objektorienterad programmering
- ▶ föreläsningar, labbar, projekt, datorbaserat tenta, basgruppsarbete
- ▶ vi kommer att använda Java genom hela höst terminen



Outline

Hemsida

Organization

Examination

Webreg

Programmering



Outline

Hemsida

Organization

Examination

Webreg

Programmering



Kursinfo: organization

<https://www.ida.liu.se/~TDDC77/material/tidsplan.ht17.pdf>

- ▶ 14 föreläsningar över HT1 och HT2: helt frivilliga, kommer att följa (ungefär) samma övergripande upplägg som de från 2016, fast med flera ändringar.
- ▶ cirka 30 laborationstillfällen för 6 obligatoriska labbar:
 - ▶ själva tillfällen är ej obligatoriska, förutom för att demonstrera laborationsuppgifter och projektet
 - ▶ fråga om hjälp, diskutera laborationer och projektuppgiften



Kursinfo: examination

Efter att ha klarat Stone (deadline 2017-09-10):

1. Sex laborationsuppgifter (HT1 och HT2). 3hp. LAB (U, G)
2. Ett programmeringsprojekt (HT2). 2hp. UPG (U,3,4,5)
3. En datorbaserat tenta. 1hp. DAT (U,3,4,5)
4. Basgruppsarbete kring vinjetter. 2hp. BAS (U,G)
 - ▶ Ett godkänt på alla moment krävs för att få ett slutbetyg
 - ▶ I så fall, bestäms slytbetygen av UPG och DAT
 - ▶ Examination är alltid individuellt,
 - ▶ Arbetet i LAB och UPG sker i grupper av två personer.



Outline

Hemsida

Organization

Examination

Webreg

Programmering



Examination: regler

- ▶ Ni får inte lämna in kopierat (även modifierat) lösningar
- ▶ Varje student måste kunna visa att hon/han förstår alla detaljer i lösningen
- ▶ Om någon form av fusk misstänks, är läraren skyldig att göra en anmälan till universitetets disciplinnämnd
- ▶ Konsekvenserna kan bli varning eller avstängning från studierna



Outline

Hemsida

Organization

Examination

Webreg

Programmering



Kursinfo: Problem?

- ▶ Frågor som rör det vi lär oss i kursen kan ställas till mig eller till laborationsassistenterna
- ▶ Övriga frågor, problem, administrativa saker osv kan man diskutera med mig
- ▶ Vill man diskutera mig så kan man prata med Ola Leifler ola.leifler@liu.se som är terminsansvarig.



Kursinfo: webreg

- ▶ “Webreg” rapportering av LAB, UPG och DAT.

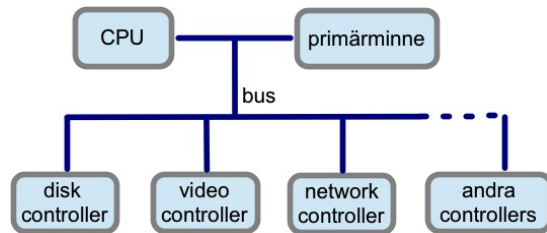
1. LAB: i grupper av två personer innan 2017-09-08,
<https://www.ida.liu.se/webreg3/TDDC77-2017-1/LAB>
2. UPG: i grupper av två personer innan 2017-10-06,
<https://www.ida.liu.se/webreg3/TDDC77-2017-1/UPG>



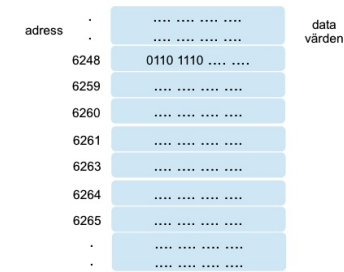
Datorer överallt ...



En "vanlig" dator ...

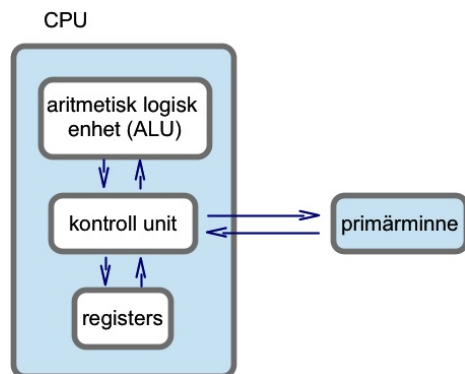


Minnet ... 8 bits ger 1 byte

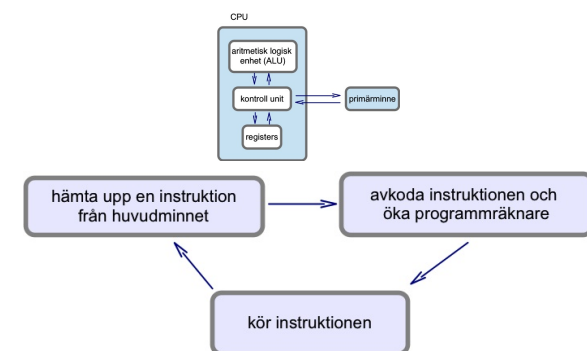


byte	B	$2^0 = 1$	10^0	B	byte
Kibibyte	KiB	$2^{10} = 1024$	10^3	KB	Kilobyte
Mebibyte	MiB	$2^{20} = 1,048,576$	10^6	MB	Megabyte
Gibibyte	GiB	$2^{30} = 1,073,741,824$	10^9	GB	Gigabyte
Tebibyte	TiB	$2^{40} = 1,099,511,627,776$	10^{12}	TB	Terabyte
Pebibyte	PiB	$2^{50} = 1,125,899,906,842,624$	10^{15}	PB	Petabyte

CPU ...



Hämta-avkoda-kör cykeln



Outline

Hemsida

Organization

Examination

Webreg

Programmering



Logga in och skapa en katalog

1. hemifrån under windows/mac/linux:
 - 1.1 du kan använda [Thinlinc](#) för att logga in på ditt konto på universitet
 - 1.2 läs mer om thinlinc på IDA genom att klicka [\[här\]](#)
2. hemifrån under windows/mac/linux:
 - 2.1 du kan också använda en "ssh" klient istället för thinlinc
 - 2.2 klicka för att läsa om hur man kan få en ssh klient under windows
 - 2.3 använd "username@remote-und.ida.liu.se" för att logga in
3. nu har du tillgång till ditt hemkatalog på IDA.
4. skapa en katalog, säg "minKod" med `mkdir minKod`
5. kom in i katalogen med `cd minKod`



Första programmet!

- ▶ i terminalen:
- ▶ skapa/öppna Hej.java i emacs med "emacs Hej.java"
- ▶ spara filen med: "C-x C-s"
- ▶ klippa/kopiera; klistra? "C-w" / "M-w"; "C-y"
- ▶ avsluta? "C-x C-c"

```
/* Programmet visa
   hur man skriva ut
*/
class Hej {

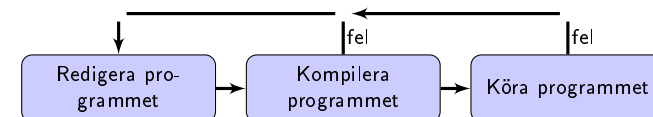
    // Skriv ut ett meddelande

    public static void main(String[] args){
        System.out.println("Hej på er TDDC77:are!!!!!!");
    }
}
```

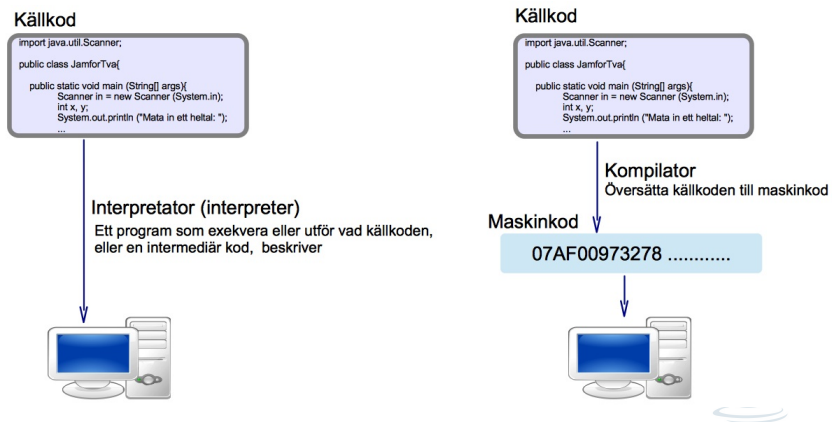


Kompilera och köra programmet

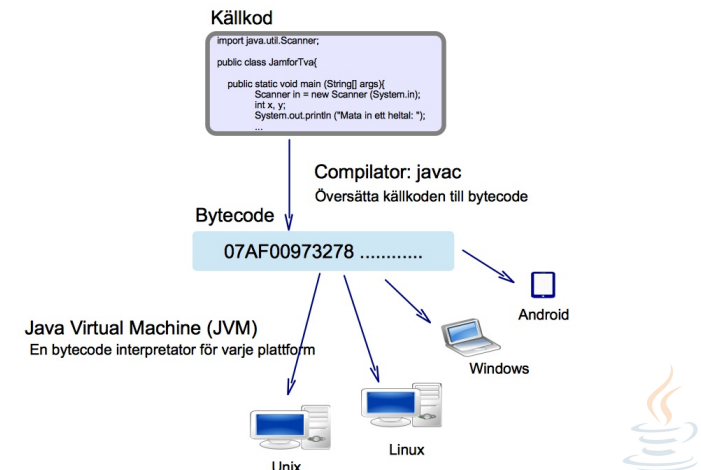
- ▶ skapa Hej.java och skriv programmet
- ▶ kompilera med "javac Hej.java"
- ▶ rätta fel och repetera tills du lyckas kompilera ditt program
- ▶ kör med "java Hej"
- ▶ syntax och semantiken



“Interpretera” eller “kompilera” källkod

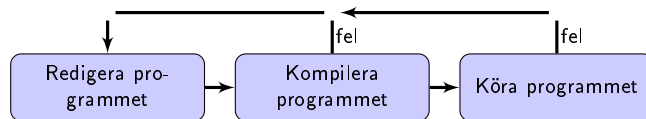


Kompilatorer och JVMer



Kompilera och köra programmet

- ▶ skapa Hej.java och skriv programmet
- ▶ kompilera med “javac Hej.java”
- ▶ rätta fel och repetera tills du lyckas kompilera ditt program
- ▶ kör med “java Hej”
- ▶ syntax och semantiken



Ge exakta instruktioner till datorn

- ▶ Exempel: hur många ord finns det i “programmering är kul” ? hur gjorde ni?
- ▶ Hur många ord i “ cd-skivan , som ligger i laptop-väskan,är trasig !” ?
- ▶ Datorer måste få exakta “instruktioner” !!

Det är viktigt att förstå att en dator gör precis vad den är sagt att göra, och INGENTING ANNAT ! Därför, måste datorer få exakta “instruktioner” !!

Exakta instruktioner

- ▶ Analysera din uppgift och försök att hitta en lösning i ett språk du kan, t.ex. svenska !
- ▶ Översätt din lösning till programkod, t.ex. Java
- ▶ Försök inte att lösa allt på en gång
- ▶ Glöm inte att:
 - ▶ Använda papper och penna
 - ▶ Prova
 - ▶ Ha kul !



Glömm inte att:

- ▶ Använda papper och penna
- ▶ Prova
- ▶ Ha kul !



Att programera är att bryta ner en uppgift i små steg som en dator kan göra

- ▶ Låt användaren skriva in två heltal x och y . Vi vill räkna upp skillnadens absoluta värde (dvs, $|x - y|$) och skriv ut resultatet.
 - ▶ Fråga användaren efter ett heltal
 - ▶ Läs in ett heltal x
 - ▶ Fråga användaren efter ett heltal
 - ▶ Läs in ett heltal y
 - ▶ Om x är större än y , räkna upp $(x - y)$
 - ▶ Om y är större än x , räkna upp $(y - x)$
 - ▶ Skriv ut resultatet

