



Programmering för Språkteknologer I

Markus Saers
markus.saers@lingfil.uu.se
9-2040
stp.ling.uu.se/~markuss/vt10/pst1



Schemaändring

- Laborationen 2010-06-01 13–15 behöver flyttas
- Mitt förslag: 2010-05-25 13–15
 - Ni har en lektion 10–12 samma dag



Kurslitteratur

- Huvudbok
 - Introduction to Programming Using Java
- Referenslitteratur
 - Java direkt med Swing
 - Webbmaterial



Lärandemål

Kursen ger grundläggande kunskaper och färdigheter i objektorienterad programmering.

Efter avslutad kurs ska studenten minst kunna:

1. på ett elementärt plan redogöra för och tillämpa en god programmeringsmetodik;



Lärandemål

2. förklara innebörden av följande begrepp och kunna skriva fungerande Javaprogram som exemplifierar och drar nytta av dem:
 - a) datatyper, objekt, klasser, variabler (instans- och klass-) och metoder (instans- och klass-);
 - b) logiska och aritmetiska operatorer, jämförelseoperatorer;
 - c) villkorsatser (if-satser) och satser för iterativ kontroll (for- och while-satser);
 - d) standardklasser, särskilt viktiga exempel på sådana t.ex. String;



Lärandemål

3. skriva program som utför elementära beräkningar utifrån förekomsten av tecken och ord-n-gram i texter;
4. använda dokumentationen av Java med viss självständighet;
5. förklara, dokumentera och kommentera kod på ett användbart sätt.

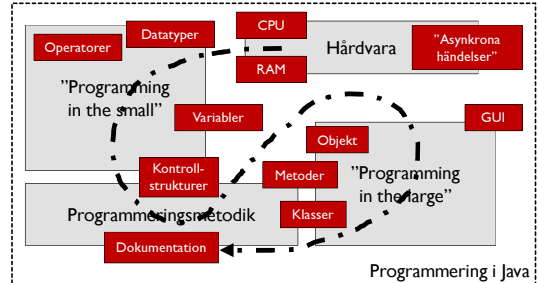


Lärandemål

- Objektorienterad programmering
- Programmeringsmetodik
- Fungerande Javaprogram
- Datatyper, objekt, klasser, variabler, metoder, operatorer, kontrollstrukturer (**if**, **for**, **while**), **String**
- Dokumentationen av Java
- Dokumentera, kommentera



Mentalt landskap



Examination

- 4 obligatoriska inlämningsuppgifter
- Tenta
- För G:
 - G på alla moment
- För VG:
 - G, samt
 - VG på 1 inlupp och tentan *eller*
 - VG på 3 inlappar



Frågor?



Hårdvara

- CPU
 - Central Processing Unit
 - Processor
- RAM
 - Random Access Memory
 - Primärminne
- Asynkrona händelser
 - Allt som kommer "utifrån"



Hårdvara

- RAM
 - Ett (stort) antal numrerade "celler"
 - Varje cell har en unik adress och kan innehålla ett heltal
- CPU
 - Innehåller ett litet antal "register"
 - Varje register fungerar som en cell
 - Kan manipulera innehållet i registren genom att utföra instruktioner
 - Ett av registren håller reda på vilken adress nästa instruktion har (program counter, PC)

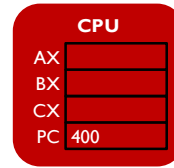


CPU och RAM

- Hämta-kör-cykeln (Fetch-and-execute cycle)
 1. Flytta en instruktion från primärminnet till processorn
 2. Utför instruktionen
 3. Bestäm var i primärminnet nästa instruktion finns
 - Vanligtvis: nästkommande instruktion
 4. Gå till 1



Exempel

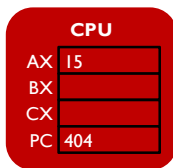


0	0
1	0
2	15
3	0
4	0
5	27
6	0
7	345

400	MOV 2 AX
404	MOV 5 BX
408	ADD
412	MOV CX 7



Exempel

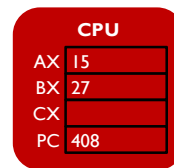


0	0
1	0
2	15
3	0
4	0
5	27
6	0
7	345

400	MOV 2 AX
404	MOV 5 BX
408	ADD
412	MOV CX 7



Exempel

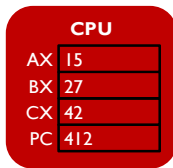


0	0
1	0
2	15
3	0
4	0
5	27
6	0
7	345

400	MOV 2 AX
404	MOV 5 BX
408	ADD
412	MOV CX 7



Exempel

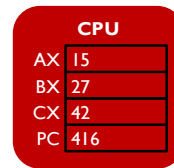


0	0
1	0
2	15
3	0
4	0
5	27
6	0
7	345

400	MOV 2 AX
404	MOV 5 BX
408	ADD
412	MOV CX 7



Exempel



0	0
1	0
2	15
3	0
4	0
5	27
6	0
7	42

400	MOV 2 AX
404	MOV 5 BX
408	ADD
412	MOV CX 7



Asynkrona händelser

- Asynkrona händelser
 - Användaren trycker på en tangent
 - Användaren använder musen
 - Den information vi ville ha från en hårddisk finns tillgänglig
 - Det information vi bad om från internet dyker upp
- Hur kan detta hanteras i en hämta-kör-cykel?
 - Polling
 1. Kolla om händelsen inträffat
 2. Om ja: hantera
 3. Om nej: gå till 1
 - Interrupts
 - När händelsen inträffar:
 1. Hantera händelsen
 2. Återuppta det som avbröts



Händelsestyrda program

- Drivs av asynkrona händelser
- Nödvändigt för grafiska användargränssnitt (GUI)
- Väl integrerad del av Java
- Om vi hinner...



Hårdvara

- Drivs av programkod på maskinspråk
 - Programmera = skriva programkod
- Varje typ av processor har ett eget maskinspråk
 - För att köra ett program på en annan dator än det är skrivet för krävs att det översätts
- Någon som vill skriva maskinkod?
 - Googla "The Art of Assembly Language Programming"
 - Långt bort från den här kursen!



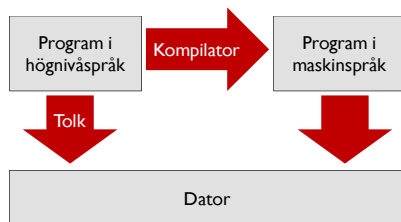
Högnivåspråk

- Basic, Pascal, C/C++, Java, Perl, Python, Ruby, ...
- Språk för att uttrycka instruktioner till datorn

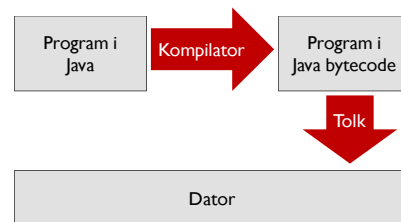

```
MOV 2 AX      (x finns i 2, y finns i 5, z finns i 7)
MOV 5 BX      z = x + y
ADD
MOV CX 7
```
- Kan översättas till maskinspråk
 - Kompilering
- Kan tolkas till maskinspråk
 - Interpretering



Kompilering/Interpretering



Kompilering/Interpretering, Java





Java konkret

- Skriv ett program
 - Skall skrivas i en fil som slutar med `.java`
- Kompilera programmet
 - `$ javac Filnamn.java`
 - Ger filen `Filnamn.class`
- Kör programmet
 - `$ java Filnamn`



Frågor?



Grundläggande programmering

- Information (data)
 - Variabler
 - Typer
- Instruktioner
 - Grundläggande instruktioner
 - Kontrollstrukturer
 - Subrutiner



Data

- Variabler
 - En minnescell som fått ett namn
 - Betydligt lättare att hantera namn än adresser!
- Typer
 - Avgör hur värdet i en minnescell ska tolkas
 - Är det ett heltal, flyttal, tecken?
- Java är "starkt typat"
 - Alla variabler måste få en typ av programmeraren



Instruktioner

- Grundläggande instruktioner
 - Operatorer (tilldelning, aritmetik, jämförelser, ...)
- Kontrollstrukturer
 - Agera olika beroende på situationen
 - Upprepa en serie instruktioner
- Subrutiner
 - Gruppera ett antal instruktioner till en enhet som kan användas upprepade gånger



Exempel

- Utgå ifrån att vi har en variabel som heter `belopp` (någonstans i ett ekonomiprogram)
- För att beräkna hur mycket beloppet ska ökas när 4 % ränta läggs till:
`ranta = belopp * 0.04;`
- För att låta räntan bero på beloppet (5 % om beloppet är över 100 000, annars 4 %):
`if (belopp > 100000)`
 `ranta = belopp * 0.05;`
 `else`
 `ranta = belopp * 0.04;`



Programmering i Java

- Programmering
 - Ge datorn instruktioner som den mekaniskt följer
- Hur ser instruktionerna ut?
 - Mitt första program™
 - Skriv ut "Hej världen!" ("Hello world!")



Ett Javaprogram

- I filen **HelloWorld.java**

```
public class HelloWorld {  
      
}  
}
```



Delarna

- I filen **HelloWorld.java** definieras en klass som heter **HelloWorld**
- Klassen **HelloWorld** innehåller en subrutin som heter **main**
 - Kallas "ingångspunkten" ("the entry point")
 - Att "köra" klassen från kommandoraden kan tolkas som att utföra subrutinen **main**
- Metoden **main** innehåller en instruktion
 - Som skriver ut texten "Hello world!"



Vad finns?

- Var kommer `System.out.println(...)` ifrån?
 - Boken, referenser på nätet, Javas API
- Grundtyper
 - Hanteras av hårdvaran
- "Komplexa" typer
 - Definieras i Java
 - Definieras av programmeraren



Grundtyper

- **byte**, **short**, **int**, **long**
 - Heltal med olika intervall
- **float**, **double**
 - Decimaltal med olika precision
- **boolean**
 - Sant/falskt
- **char**
 - 16 bitars Unicode tecken



Variabler

- Har ett namn
- Har en typ
- Innehåller ett värde
- Finns efter att de har deklarerats
- Exempel

```
int amount;  
int interest;
```



Operatorer

- Förändrar och kombinerar värden av grundtyperna
- I princip "intuitiva"!

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```



Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```




Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```

amount	10000



Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```

amount	10000
interest	



Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```

amount	10000
interest	



Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```

amount	10000
interest	400



Vad händer i datorn?

```
public class InterestTest {
    public static void main(String[] args) {
        int amount = 10000;
        int interest;
        if (amount > 100000)
            interest = amount * 0.05;
        else
            interest = amount * 0.04;
    }
}
```

amount	10000
interest	400

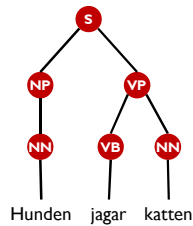


Operatörer

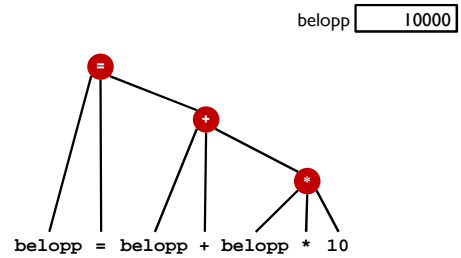
- Aritmetiska: + - / * %
- Jämförande: == != < > <= >=
- Booleska: ! && ||
- Tilldelning: =
- Öka/minska: ++ --
- Kombinerade
 - `belopp = belopp + 1000;`
 - `belopp += 1000;`



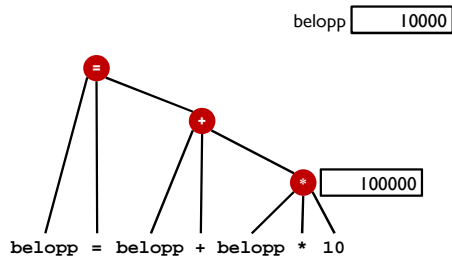
Syntax



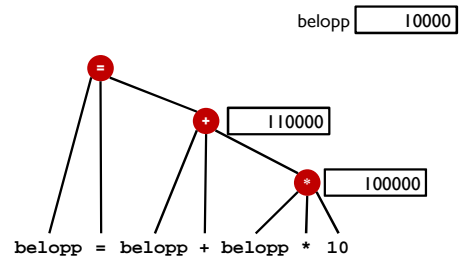
Syntax



Syntax



Syntax



UPPSALA UNIVERSITET **Syntax**

belopp

belopp = belopp + belopp * 10

UPPSALA UNIVERSITET **Syntax**

belopp

belopp = (belopp + belopp) * 10

UPPSALA UNIVERSITET **Syntax**

belopp

belopp = (belopp + belopp) * 10

UPPSALA UNIVERSITET **Frågor?**

UPPSALA UNIVERSITET **I/O**
Input/Output

- I **shared** mappen för kursen finns **TextIO.java** som förekommer i boken
- Se till att checka ut **shared** mappen, och kopiera **TextIO.java** till din arbetskatalog
- Klassen **TextIO** innehåller en massa subrutiner som kan användas för att läsa in värden från användaren

UPPSALA UNIVERSITET **Exempel**

```

public class Test {
    public static void main(String[] args) {
        int ettHeltal;
        System.out.print("Skriv in ett heltal: ")
        ettHeltal = TextIO.getInt();
        System.out.print("Du skrev in ");
        System.out.println(ettHeltal);
    }
}

$ javac Test.java
$ java Test
Skriv in ett heltal: 12
Du skrev in 12
$

```



TextIO

- `getBoolean()`
- `getByte()`
- `getChar()`
- `getDouble()`
- `getFloat()`
- `getInt()`



Strängar

- En sträng är en serie med tecken ("Hej världen!")
- Väldigt användbart!
- Inte en grundläggande typ, utan en standardklass
 - `String`



Exempel

```
public class Test {
    public static void main(String[] args) {
        String name;
        System.out.print("Vad heter du? ")
        name = TextIO.getln();
        System.out.print("Hej ");
        System.out.println(name);
        System.out.println(",");
        System.out.print("Jag sa: \"HEJ ");
        System.out.print(name.toUpperCase());
        System.out.println("!\".");
    }
}
```