

OOMPA 2001

Föreläsning 1

Introduktion till Java del a

***Grunder: Javas uppbyggnad, konstruera klass,
kompilera, köra, lite syntax***

Något om IDEer: JBuilder

Av Björn Eiderbäck
Email:
bjorne@nada.kth.se
Adress:
Rum 1641, 6tr
NADA
Osquars Backe 2
Tel: 7906277

◀ previous next ▶

DOMPAE-010903 Introduktion Java, del a

Momentets upplägning och innehåll...

- Föreläsningar och övningar: introduktion till Java
 - På kursen 2D1360 ingår två introduktionsföreläsningar och en övning om och med Java. Resten av kursen samläses med 2D1359.
 - Måndag 3/9 kl 10-12
 - Java introduktion.
 - Måndag 3/9 kl 13-15
 - Mer om Java, framförallt via exempel.
 - Tisdag 4/9 kl 15-17
 - Första föreläsningen tillsammans med 2D1359.
 - Onsdag 5/9 kl 8-10
 - Föreläsning tillsammans med 2D1359.
 - Onsdag 5/9 kl 10-12
 - Förberedelse inför introduktionslab (ej obligatorisk) i Java
 - Onsdag 5/9 kl 13-15
 - Introduktionslab (ej obligatorisk) i Java

◀ previous next ▶ 2

DOMPAE-010903 Introduktion Java, del a

Java

- Java är konstruerat på SUN
 - Första versionen släpptes 1995
- Starkt typat
 - Dvs det går att avgöra typkompatibilitet av alla uttryck i programmet vid kompileringstillfället.
- Virtuellt maskin (VM) som gör det maskinberoende
 - En speciell VM per plattformstyp överbryggar olikheter i plattformarnas arkitektur
- Klassbibliotek
 - språkklasser, stränghantering, grafik, gränssnitt, fönster, Internet, händelsehantering, processer, mängder, mm
- Bindning till WWW
 - med applets och liknande
- Har fått sin huvudsakliga spridning via Internet
- Syntax som C. Semantik och klasser mer som Smalltalk

◀ previous next ▶

3

DOMPAE-010903 Introduktion Java, del a

Virtuell maskin

```
graph TD; K[källkod] -- "kompilera (javac)" --> B[/java bytekod/]; B -- "kör (java)" --> P[Pc VM]; B -- "kör (java)" --> M[Mac VM]; B -- "kör (java)" --> U[Unix VM];
```

en virtuell maskin per plattformstyp

◀ previous next ▶

4

DOMPAE-010903 Introduktion Java, del a

Java: mitt första "program"

- Skapa en fil med namnet `MyFirstProgram.java`
- Skriv följande kod i filen:

```
import java.io.*;

public class MyFirstProgram{
    public static void main (String [] args){
        System.out.println("Mitt första program!");
    }
}
```

◀ previous | next ▶ 5

DOMPAE-010903 Introduktion Java, del a

... kompilera och kör

- Kompilera
- Då genereras en körbar fil `MyFirstProgram.class`
- Kör genom att starta Javas virtuella maskin med program/klassnamnet som argument
- Regeln är att klassens *static void main-metod* alltid exekveras då man startar den från terminal på detta sätt

◀ previous | next ▶ 6

DOMPAE-010903 Introduktion Java, del a

Ett "Javaprograms" struktur

- En klass ser ut på följande sätt


```
public class KlassNamn extends
    SuperKlassNamn{
    attribut (instans- och klassvariabler)
    metoder (instans- och klassmetoder)
    konstruktörer
    }
```

Kallas också för medlemmar
- **En metod**

```
modifierare returtyp funktionsnamn(argument){
    programsatser
}
```

◀ previous | next ▶

7

DOMPAE-010903 Introduktion Java, del a

Konstruera klass

- Klassnamn = filnamn
 - En klass skall ha samma namn som den fil den placeras i
 - Klassen Circle i filen Circle.java
 - En fil kan innehålla flera klasser men bara en av dem kan vara deklarerad public,
 - Det är den publika klassens namn som motsvarar filens
- Definition av klass, mall


```
public class KlassNamn extends SuperKlassNamn{
    Modifierare TypIvar1 ivar1;
    Modifierare MetodTyp metod1(TypArg1 arg1) {}
    public KlassNamn() {}
}
```

instansvariabler

metoder

konstruktörer

◀ previous | next ▶

8

DOMPAE-010903 Introduktion Java, del a

Deklarera och instansiera

- Deklaration av instans
`KlassNamn variabelNamn;`
- Instansiering
`variabelNamn = new KlassNamn();`

alternativt, både deklaration och instansiering på en gång

```
KlassNamn variabelNamn = new KlassNamn();
```

◀ previous | next ▶ 9

DOMPAE-010903 Introduktion Java, del a

Kompilera och köra

- Environmentvariabel
 - För bekväm hantering sätt path (se systemdokumentation) så att kompilator, virtuell maskin och bibliotek enkelt hittas
 - Plattformsberoende hur detta går till
- Kompilera med javac
`javac KlassNamn.java`
 - om allt går bra skapas då en fil med namnet
`KlassNamn.class`
- Kör med java
`java KlassNamn`

◀ previous | next ▶ 10

OMP AE-010903 Introduktion Java, del a

Sammanfattning enkelt program: Hello World

Det mesta i Java tillhör klasser

Alla program måste ha en main()

```

public class Hello {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
    
```

Spara i en fil med namnet Hello.java

Kompilera med javac Hello.java

Då skapas en binärfil Hello.class

Kör med java Hello

◀ previous
next ▶
11

OMP AE-010903 Introduktion Java, del a

Typer

- I Java måste alla variabler och metoder deklarerars med typ
Typ variabel;
- Det finns primitiva typer som int, char, float och boolean
int x;
- En variabel kan ges ett initialt värde vid deklarationen
int x = 10;
- Man kan också använda typer som är implementerade som vanliga klasser
String course = "OOMPA-01";

◀ previous
next ▶
12

DOMPAE-010903 Introduktion Java, del a

Modifierare

- En modifierare kan användas för att ange vilka objekt som kan accessa en klass eller dess medlemmar (dvs dess attribut eller metoder), mest grundläggande är:
 - `static`
 - indikerar att medlemmen är klassvariabel eller klassmetod
 - `public`
 - medlemmen är accessbar överallt där dess klass är accessbar
 - `protected`
 - medlemmen endast accessbar i klassen, dess subklasser eller i aktuellt package (vi återkommer till package vid senare tillfälle)
 - `private`
 - medlemmen endast accessbar i den klass som definierar den
 - accessbar av andra klasser i samma paket

◀ previous | next ▶ 13

DOMPAE-010903 Introduktion Java, del a

Exempel

```
public class BankAccount{
    protected int balance = 0;

    public void deposit (int amount) {
        balance = balance + amount;
    }

    public void withdraw (int amount) {
        balance = balance - amount;
    }

    public int balance (){
        return balance;
    }
}
```

◀ previous | next ▶ 14

DOMPAE-010903 Introduktion Java, del a

```
...  
BankAccount account1, account2;  
int transfer;  
  
account1 = new BankAccount();  
account2 = new BankAccount();  
  
account1.deposit(200);  
account2.deposit(100);  
  
transfer = 75;  
account1.withdraw(transfer);  
account2.deposit(transfer);  
account2.balance(); // resultat 175
```

◀ previous | next ▶

15

DOMPAE-010903 Introduktion Java, del a

Exempel: Klass

```
// Klassdefinition (utan extends medför subklass till Object)  
public class Circle {  
    public double x, y;  
    public double r;  
    public double circumference(){  
        return 2 * 3.14 * r;  
    }  
    public double area(){  
        return 3.14 * r * r;  
    }  
  
    // för att testa det hela skriver vi en static main-metod  
    public static void main (String args[]) {  
        //Vi deklarerar en temporär variabel  
        Circle circle;  
        // och instansierar  
        circle = new Circle();  
    }  
}
```

◀ previous | next ▶

16

DOMPAE-010903 Introduktion Java, del a

... forts ...

```
//vi ändrar några värden
circle.x = 100; circle.y = 200; circle.r = 20;
// vi gör några testutskrifter
System.out.println("Cirkelns x-koordinat: " +
    circle.x + " och dess y-koordinat: "
    + circle.y);
System.out.println("den har en radie också: " +
    circle.r);
System.out.println(
    "metoden circumference ger omkretsen: " +
    circle.circumference());
System.out.println("och med area:" + circle.area() +
    " dess area");
}
}
```

◀ previous | next ▶ 17

DOMPAE-010903 Introduktion Java, del a

... och så sparar vi, kompilarar och kör

- Spara
 - Spara filen med samma namn som klassen med *extension* **.java**, dvs här `Circle.java`
- Kompilera
 - `javac Circle.java`
 - Skapar en fil `Circle.class`
- Kör
 - `java Circle`

◀ previous | next ▶ 18

DOMPAE-010903 Introduktion Java, del a

Exempel: Eka argument

- Skapa fil Echo.java (som ekar argumenten vid exekveringen)


```
public class Echo {
    public static void main(String[] argv) {
        for(int i = 0; i < argv.length; i++)
            System.out.print(argv[i] + " ");
        System.out.print("\n");
    }
}
```

forloopar ser ut som i ANSI-C

- Kompilera


```
javac Echo.java
```
- Kör


```
java Echo detta är input
```

◀ previous | next ▶ 19

DOMPAE-010903 Introduktion Java, del a

In- och utmatning

- Textutmatning sker via klassen System


```
System.out.println(...);
```

 - Där:
 - System är en klass i java.lang
 - out är en klassvariabel i System
 - out är en instans av klassen `PrintStream`
 - `PrintStream` har en instansmetod `println` som skriver ut text på terminalen
- Textinmatning sker också via klassen System


```
char c = System.in.read();
```

 - Inmatningen måste dock "fånga" undantag


```
{char c = System.in.read()
    (IOException e){//fel har inträffat}
```

◀ previous | next ▶ 20

DOMPAE-010903 Introduktion Java, del a

Exempel: "program" med teckeninmatning

Här använder vi paket också!

```

package Tests;
import java.io.*;
class TestIO_1_1 {
    public static void main(String [] args) {
        for(;;){
            int in;
            try
                {in = System.in.read();}
            catch(IOException e)
                {in = -1;}
            if ((char) in == '0') return;
            System.out.println(in);
        }
    }
}
    
```

21

DOMPAE-010903 Introduktion Java, del a

Exempel: dialog med stränginmatning

```

package Tests;
import java.io.*;
class TestIO {
    public static void main(String [] args) throws IOException
    { System.out.println("TEST OF INTERACTION/1");
      BufferedReader in = new BufferedReader(
          new InputStreamReader(System.in));

      for(;;) {
          System.out.print(">");
          String line = in.readLine();
          if (line.equalsIgnoreCase("quit"))
              break;
          if (line.equalsIgnoreCase("help"))
              System.out.println("To quit: write 'quit'");
          else System.out.println("\teko: " + line);
      }
    }
}
    
```

22

DOMPAE-010903 Introduktion Java, del a

Java: läsa argumenten som ges via terminalen

```
public class ShowArgs {
    public static void main(String[] args) {
        for(int i=0; i<args.length; i++)
            System.out.println("Arg " + i + " is "
                               + args[i]);
    }
}
```

Testa med tex

```
java ShowArgs arg1 arg2 arg3
```

◀ previous | next ▶

23

DOMPAE-010903 Introduktion Java, del a

... “använd” syntax

- I föregående exempel använde vi en for-loop
 - Syntaxen är följande:


```
for(initialvärde; villkor; uppdatera)
    sats
```
- Andra loopkonstruktioner
 - while-loop


```
while(villkor) sats
```
 - do-loop


```
do sats while(villkor)
```
- Där en sats antingen är
 - enstaka sats som `System.out.println("Test");`
 - eller flera satser inom måsvingar som


```
{ x += 10;
  System.out.println("X: " + x);
}
```

◀ previous | next ▶

24

DOMPAE-010903 Introduktion Java, del a

En IDE

- En IDE är en Integrated Development Environment som är en hel miljö som kan ge stöd vid programutveckling
- Vi tittar kort på JBuilder, som finns (gratis) för både Unix, tex Sun Solaris, och Windows

◀ previous | next ▶

25

DOMPAE-010903 Introduktion Java, del a

Java: exempel med bla villkor och modulo

är modulo

```

public class FizzBuzz {
    public static void main(String[] args) {
        for(int i = 1; i <= 100; i++) {
            if ((i % 5) == 0) && ((i % 7) == 0))
                System.out.print("fizzbuzz");
            else if ((i % 5) == 0)
                System.out.print("fizz");
            else if ((i % 7) == 0)
                System.out.print("buzz");
            else
                System.out.print(i);
            System.out.print(" ");
        }
        System.out.println();
    }
}
                
```

◀ previous | next ▶

26

DOMPAE-010903 Introduktion Java, del a

Villkor

- Villkor kan skrivas

```
if (uttryck) sats
```

 - **eller**

```
if (uttryck)
    sats
else sats
```
- Man kan ibland också använda ?:
`uttryck ? omSantVärde : annarsVärde`
- Eventuellt switchar

```
switch(heltal) {
    case värde1 : sats1;
                    break;

    case värde2 : sats2;
                    break;

    ...
    default: satsN;
}
```

◀ previous | next ▶ 27

DOMPAE-010903 Introduktion Java, del a

Java: fler exempel, Fibonacci

```
public class Fibonacci {
    public static void main(String[] args){
        int current, prev = 1, prevprev = 0;
        for(int i = 0; i < 20; i++) {
            current = prev + prevprev;
            System.out.print(current + " ");
            prevprev = prev;
            prev = current;
        }
        System.out.println();
    }
}
```

◀ previous | next ▶ 28

DOMPAE-010903 Introduktion Java, del a

Java: ännu fler exempel

```
public class Reverse {
    public static void main(String[] args) {
        // Loopa baklänges genom orden som ges som argument
        for(int i = args.length-1; i >= 0; i--) {
            // Loopa baklänges genom tecknen i varje argument
            for(int j=args[i].length()-1; j>=0; j--) {
                // Skriv ut tecken j i i:te argumentet.
                System.out.print(args[i].charAt(j));
            }
            System.out.print(" ");
        }
        System.out.println();
    }
}
```

29

◀ previous | next ▶

DOMPAE-010903 Introduktion Java, del a

Java: exempel, iterativ fakultetsfunktion

```
public class Factorial {
    public static int factorial(int x) {
        int fact = 1;
        for(int i = 2; i <= x; i++)
            fact *= i;
        return fact;
    }
}
```

30

◀ previous | next ▶

DOMPAE-010903 Introduktion Java, del a

Java: exempel, rekursiv fakultetsfunktion

```
public class Factorial2 {  
    public static long factorial(long x) {  
        if (x == 1) return 1;  
        else return x * factorial(x-1);  
    }  
}
```

◀ previous next ▶

31