

Kort om **java** eller **Java** för den som kan **C++**

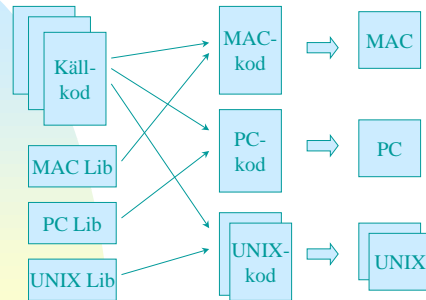
- Historik
- Java Virtual Machine
- Varför Java? Varför inte?
- Java och C++
- Språket Java
- Apps och Applets
- Wizards, events, threading, animering

2000-12-09

Thomas Johansson, Datavetenskap

1

Konventionella program

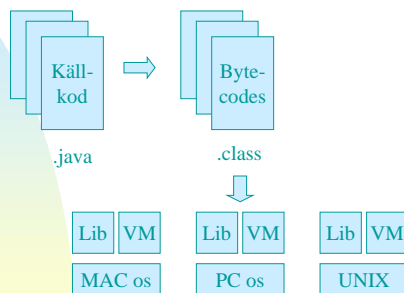


2000-12-09

Thomas Johansson, Datavetenskap

2

Java Virtual Machine



2000-12-09

Thomas Johansson, Datavetenskap

3

Varför java? Varför inte?

- Maskinoberoende
 - ◆ Virtuellt maskin
- Säkerhet
 - ◆ Ingen direkt tillgång till systemet
- Storlek
- Prestanda
 - ◆ Just-in-time

2000-12-09

Thomas Johansson, Datavetenskap

4

Olika typer av program

- Applications (Apps)
 - ◆ 'vanliga' program
- Applets
 - ◆ Bytecodes i Web-sidor
- Javascript ≠ Java
 - ◆ Källkod direkt i Web-sidor

2000-12-09

Thomas Johansson, Datavetenskap

5

Java och C++

- Bygger mycket på C++
 - ◆ Inget äkta superset av C eller C++
- Rensat upp i C/C++
 - ◆ inga pekare
 - ◆ inga variabla argumentlistor (a la printf)
 - ◆ ingen preprocessor
 - ◆ inga enum
- Garbage collection

2000-12-09

Thomas Johansson, Datavetenskap

6

Ett litet Java-program

```
public class Appl
{
    public static void main(string args[])
    {
        system.out.println("Hej Hopp");
    }
}
```

2000-12-09

Thomas Johansson, Datavetenskap

7

Datatyper

- boolean 8 bitar
- byte 8
- char 16 (Unicode)
- short 16
- int 32
- float 32
- double 64
- string
- Casting är nödvändigt ibland

2000-12-09

Thomas Johansson, Datavetenskap

8

Strängar

- Typen string
- Ej '\0'-avslutad som i C
- 16 bitars Unicode
- Konkaterering: "Värdet är " + v + " kr."

2000-12-09

Thomas Johansson, Datavetenskap

9

Kontrollstrukturer

- Jämförelser ger Boolean som resultat
 - ◆ har inte med heltal att göra
- If
- While
 - ◆ kan ha labels för uthopp ur nästade loopar
- For
- Switch

2000-12-09

Thomas Johansson, Datavetenskap

10

Klasser

- En klass per .java - fil
- Enda sättet att skapa en ny typ
- Alla funktioner måste vara medlemmar
 - ◆ Default returtyp är Void
 - ◆ Inga argument -> tom parameterlista
- Alla funktioner måste deklaras i klassen
- Final, final static

2000-12-09

Thomas Johansson, Datavetenskap

11

Exempel

```
class BankAccount
{
    double balance

    void deposit(double amount)
    {
        if(amount >= 0.0)
            balance = balance + amount
    }

    double getBalance()
    {
        return balance;
    }
}
```

2000-12-09

Thomas Johansson, Datavetenskap

12

'Access control'

- Default är friendly
 - ◆ åtkomst inom samma 'package'
- public
 - ◆ åtkomst från alla paket
- protected
 - ◆ åtkomst för 'arvingar'
- private
 - ◆ åtkomst endast inom samma klass

2000-12-09

Thomas Johansson, Datavetenskap

13

Referenser

- Inga pekare, referenser i stället
- Garbage collection
 - ◆ reference counts
- Arrays
 - ◆ int vektor[]: ingen storlek tillåten!
 - ◆ Använd **new**
 - ◆ Koll av index
 - ◆ array of array: `int matrix = new int[2][];`
 - ◆ kolumnerna inte lika långa

2000-12-09

Thomas Johansson, Datavetenskap

14

Konstruktörer

- Variabler initieras automatiskt till 0 eller false
- Copy constructor behövs sällan
- Finalizer
 - ◆ anropas när objektet återlämnas, inte när det försvinner
 - ◆ låt garbage collection göra jobbet!

2000-12-09

Thomas Johansson, Datavetenskap

15

Arv

- Extends
 - ◆ `class CheckingAccount extends BankAccount`
- interface
 - ◆ något en klass måste implementera
 - ◆ 'lättniktstyp'
- implements
 - ◆ inte ett riktigt arv

2000-12-09

Thomas Johansson, Datavetenskap

16

Packages

- En grupp av klasser
 - ◆ `package namn1.namn2...`
 - ◆ `import namn1`
 - ◆ `unnamed`
- Windows lagrar klasserna i olika bibliotek

2000-12-09

Thomas Johansson, Datavetenskap

17

- `srd\math\ComplexNumber.class`

```
// deklaration
package srd.math
public class ComplexNumber { ... }

// användning
srd.math.ComplexNumber c1;

// eller
import srd.math
ComplexNumber c1;
```

2000-12-09

Thomas Johansson, Datavetenskap

18

Java class library

- java.lang
 - ◆ fundamentala klasser
 - ◆ Object
 - ◆ String
 - ◆ Class Wrappers
 - ◆ automatiskt importerad
- java.applet
 - ◆ www applets
 - ◆ trådning
 - ◆ händelsestyrning

2000-12-09

Thomas Johansson, Datavetenskap

19

- java.awt
 - ◆ maskinberoende fönsterverktyg
- java.io
 - ◆ filhantering
- java.net
 - ◆ internet i/o
- java.util
 - ◆ blandat, exempelvis
 - ◆ datum

2000-12-09

Thomas Johansson, Datavetenskap

20

Applets

- Används i web-sidor
- alla klasser ärver från **Applet**
- Funktioner:
 - ◆ `init()` anropas när en applet laddas
 - ◆ `destroy()` sidan försvinner ur minnet
 - ◆ `paint()` rita om fönstret
 - ◆ `start()` när browsern visar fönstret
 - ◆ `stop()` när browsern lämnar sidan

2000-12-09

Thomas Johansson, Datavetenskap

21

Events

- När något händer
 - ◆ `mouseMove(Event, int, int)`
 - ◆ `mouseDrag(Event, int, int)`
 - ◆ `mouseDown(Event, int, int)`
 - ◆ `keyDown(Event, int)`
 - ◆ ...

2000-12-09

Thomas Johansson, Datavetenskap

22

Multithreading

- Flera 'trådar' samtidigt
- En slags lättviktsprocesser
 - ◆ delar samma minne

2000-12-09

Thomas Johansson, Datavetenskap

23

```
class MyClass implements Runnable
{
    Thread m_thread;

    public void run()
    {
        // här kan det vara en oändlig loop
    }

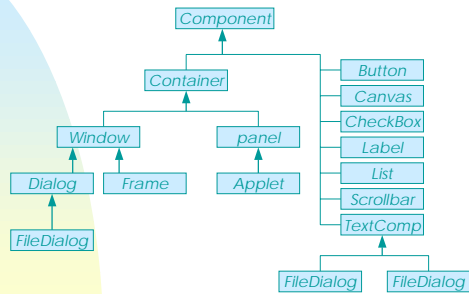
    void startNewThread()
    {
        m_thread = new Thread(this);
        m_thread.start();
    }
}
```

2000-12-09

Thomas Johansson, Datavetenskap

24

Abstract Windowing Toolkit



2000-12-09

Thomas Johansson, Datavetenskap

25

Java Beans

- Komponenter som kan manipuleras vid designen
 - ◆ knappar
 - ◆ listor
 - ◆ 'radioknappar'
- Motsvarar en klass i AWT
- jmf VBX, OCX i Visual Basic, Delphi

2000-12-09

Thomas Johansson, Datavetenskap

26

Java på internet

- www.javasoft.com/docs/books/jls/html/index.html
- www.javasoft.com/Series/Tutorial/book.html
- www.javasoft.com/docs/books/vmspec/
- www.microsoft.com/visualj/
- www.gamelan.com

2000-12-09

Thomas Johansson, Datavetenskap

27