




UPPSALA
UNIVERSITET

Programmering för Språkteknologer I


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



UPPSALA
UNIVERSITET

Frågor?


- På boken, något som är oklart?
- På övningarna, varför måste man göra såhär?
- På inlämningsuppgifterna (-en), kanske inte än?



UPPSALA
UNIVERSITET

Grundläggande typer


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



UPPSALA
UNIVERSITET

Heltal i datorer


- All information kodas med bitar
- En bit är en binär siffra (1 eller 0)
 - Eller på/av, sant/falskt, ...
- Ett tal representeras med ett antal bitar
- Antalet olika tal som kan representeras beror på antalet bitar som används
 - 8 bitar = $2^8 = 256$ olika tal (0–255)
 - Varje extra bit dubblar möjligheterna



UPPSALA
UNIVERSITET

Heltal


- `byte`
 - Grundläggande enhet i datalagring (b, kb, Mb, Gb, ...)
 - 8 bitar: -128 till 127
- `short`
 - 16 bitar: -32 768 till 32 767
- `int`
 - 32 bitar: -2 147 483 648 till 2 147 483 647
- `long`
 - 64 bitar: -9 223 372 036 854 775 808 till 9 223 372 036 854 775 807



UPPSALA
UNIVERSITET

Decimaltal


- Lagras ungefärligt i datorn
- `float`
 - 32 bitar
 - Ungefär 7 värdesiffror
 - $32.39892311134 == 32.3989234399 == 32.398923$
- `double`
 - 64 bitar
 - Ungefär 15 värdesiffror



UPPSALA
UNIVERSITET

Boolean


- En bit information
 - Sant/falskt
- Kan i praktiken kräva större utrymme i datorn



UPPSALA
UNIVERSITET

Tecken


- Ett tecken från tecknetabellen UTF-16
- 16 bitar (65 535 olika tecken)



UPPSALA
UNIVERSITET

"Literals"


- Kallas ibland konstanter på svenska
- Decimaltal
 - Måste anges i decimalform (12.0)
 - Typen float behöver ett f efter sig (12.0f)
- boolean
 - true false
- char
 - Tecken omslutna med enkelt citattecken ('a')
- String
 - Noll eller fler tecken omslutna med citattecken ("Hello world")



UPPSALA
UNIVERSITET

Heltals-"literals"


- Decimalform
 - Som vanligt!
 - En long behöver ett L efter sig
- Oktalform
 - Bara siffrorna 0-7
 - Inleds med 0
 - 045 = 37
- Hexadecimalform
 - "Siffrorna" 0-9, a, b, c, d, e, f
 - Inleds med 0x
 - 0x20 = 32, 0x0e = 14



UPPSALA
UNIVERSITET

Specialtecken

- Tecken som inleds med bakstreck ("back slash") behandlas annorlunda
- "\" en strängkonstant som innehåller ett citattecken
- Kontrolltecken till skrivare (\n\r)
- Tecknet bakstreck (\)
- Namngivna unicode tecken: \uXXXX
 - Där XXXX är tecknets id i hexadecimalform (utan 0x)
 - "\u004d" kan också skrivas m
 - '\u00f8' (ett norskt ö)



UPPSALA
UNIVERSITET

Frågor?

UPPSALA
UNIVERSITET

Kommentarer

- Ignoreras av Java
- Tre varianter
 - `//`
 - Allt efter detta på samma rad ignoreras
 - `/* */`
 - Allt mellan dessa ignoreras
 - `/** */`
 - Allt mellan dessa ignoreras av `javac`, men kan användas för att bygga dokumentationssidor

UPPSALA
UNIVERSITET

Klasser

- Klasser kan användas för tre ändamål
- För att representera saker (objekt) av en viss typ
 - `String`
- För att samla ett antal subrutiner som hör ihop
 - `Math`
 - `(System)`
- För att köra som program
 - `HelloWorld`

UPPSALA
UNIVERSITET

Användningsområden

```
String s = "en sträng";
System.out.println(s.toUpperCase());

int i = (int)(Math.random() * 6) + 1;

$ java HelloWorld
```

UPPSALA
UNIVERSITET

Subrutiner/Funktioner/Metoder

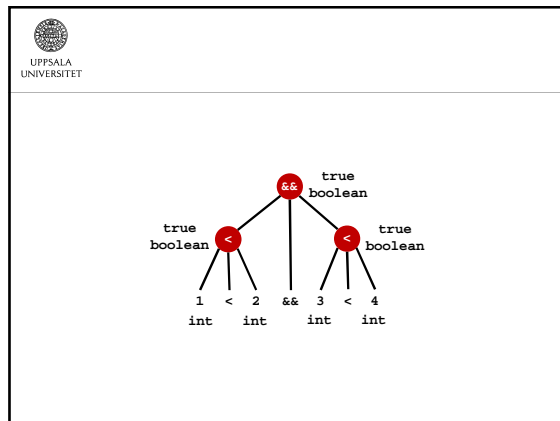
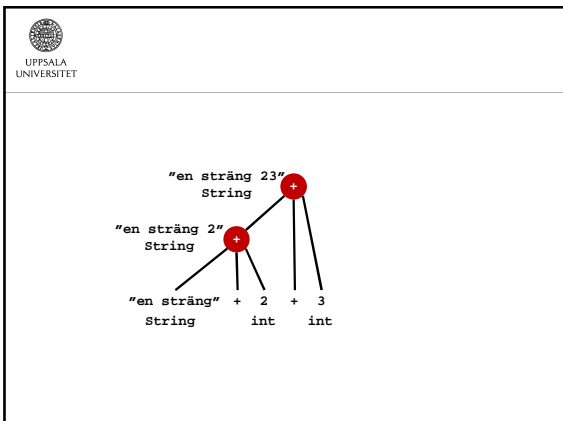
- En subrutin utför en förbestämd uppgift
 - `System.out.println()`
- En funktion är en subrutin som ger ett resultat
 - `Math.random()`
- En metod är en subrutin eller funktion som hör till en klass (formellt sett alla subrutiner/funktioner i Java)
 - `s.toUpperCase()`

UPPSALA
UNIVERSITET

Resultat från operatörer

- Aritmetiska uttryck
 - Inlämningsuppgift 1
 - Går att konkatenera saker med `+`
 - "summan blev" + `d1` + `d2`
- Jämförelseoperatörer och booleska operatörer
 - Resulterar i en boolean

UPPSALA
UNIVERSITET



UPPSALA UNIVERSITET

Frågor?

UPPSALA UNIVERSITET

Uppräkningar

- "Enumerations" på engelska
- Bra sätt att hantera kategoriska data exempelvis veckodagar
- Istället för att ha ett heltal 1-7 för olika dagar kan man räkna upp dagarna istället (enklare att hantera)

```
enum Dag {
    MANDAG, TISDAG, ONSDAG, TORSDAG,
    FREDAG, LORDAG, SONDAG
}
```

- Dag är nu en typ, som kan ta värdena MANDAG...SONDAG

```
Dag d = Dag.TISDAG;
```

UPPSALA UNIVERSITET

Inkrementering / Dekrementering


- "Öka med ett"/"Minska med ett"
- Finns i två versioner: prefix och postfix
- Prefix: uttrycket har samma värde som det förändrade värdet
- Postfix: uttrycket har samma värde som det oförändrade värdet

UPPSALA UNIVERSITET

Exempel


```
public class Test {
    public static void main(String[] args) {
        int i = 10;
        System.out.println(++i);
        System.out.println(i++);
        System.out.println(i);
        System.out.println(i--);
        System.out.println(--i);
    }
}
```

- Vad skrivs ut, och varför?



"Typecast"


- Tvinga fram en tolkning av ett värde
- Skriv namnet på typen inom parenteser framför det som ska tolkas om



Exempel


```
int i = 15;
System.out.println( i / 2 );

int i = 15;
System.out.println( (double)i / 2 );
```




Konvertering till och från String

- Använd klasserna Integer (för int) och Double (för double)
 - double Double.parseDouble(String)
 - String Double.toString(double)
 - int Integer.parseInt(String)
 - String Integer.toString(int)
- Finns liknande för byte, short, long, float och boolean
- Inte för char, eftersom String är en serie med tecken




Frågor?



Block och variabler


- Ett block består av ett antal satser inneslutna mellan krullparenteser/måsvingar


```
{
  // Allt
  // det här
  // tillhör
  // samma block
}
```
- Variabler "lever" inte utanför den låda de deklarerats i
 - Variabler som skapas i main-metoden finns bara där, och "dör" när main-metoden körts färdigt



Kontrollstrukturer

- Loopar (Iterationssatser)
 - Upprepar ett block flera gånger
- Konditionalsatser
 - Väljer vilket block som ska köras




UPPSALA
UNIVERSITET

while

- **Form:**

```
while (villkor) sats;
while (villkor) block
```
- **Funktion:**
 Utför *sats/block* så länge *villkor* är uppfyllt
- **Exempel:**

```
int i = 10;
while (i != 0) {
    System.out.println(i);
    i -= 1;
}
```




UPPSALA
UNIVERSITET

if (med satser)

- **Form:**

```
if (villkor)
    sats1;
else
    sats2;
```
- **Funktion:**
 Om villkor är sant körs *sats1*, annars körs *sats2*
- **Exempel:**

```
int i = 10;
if (i % 2 == 0)
    System.out.println(i + " är jämnt");
else
    System.out.println(i + " är ojämnt");
```




UPPSALA
UNIVERSITET

if (med block)

- **Form:**

```
if (villkor) block1
else block2
```
- **Funktion:**
 Om villkor är sant körs *block1*, annars körs *block2*
- **Exempel:**

```
int i = 10;
if (i % 2 == 0) {
    System.out.println(i + " är jämnt");
} else {
    System.out.println(i + " är ojämnt");
}
```




UPPSALA
UNIVERSITET

do ... while

- **Form:**

```
do block while (villkor);
```
- **Funktion:**
 Utvärdera villkor efter första iterationen
- **Exempel:**

```
int i = 0;
do {
    System.out.println(i);
} while (i > 0);
```



UPPSALA
UNIVERSITET

for

- **Vanlig förekommande while-loop:**

```
int i = 0;
while (i < 10) {
    // Gör något med i
    i += 1;
}
```
- **Kan förkortas till en for-loop:**

```
for (int i = 0; i < 10; i += 1) {
    // Gör något med i
}
```