

Objektorienterad Programmering (TDDC77)

Föreläsning III: uttryck, interaktiva program, API:et

Ahmed Rezine

IDA, Linköpings Universitet

Hösttermin 2016



Aritmetiska uttryck

- ▶ Ett uttryck är en kombination av en eller mer operatörer (operators) och operander (operands).
- ▶ En operatör kan vara unär, binär, etc
- ▶ En operand kan vara en litteral (-3, 0, 10.5, etc) en konstant eller en variabel
- ▶ Används vanligtvis för att utföra en beräkning och resulterar i ett antal (fast inte nödvändigtvis)

```
3+4.5  
18-(12/2)  
''resultatet blir:'' + (10\%(5/3))
```



Aritmetiska uttryck

Definierat för heltals och flyttalstyper:

- ▶ Addition +, Subtraktion -, Multiplikation *, Division /, Resten %. t.ex. 18 % 5 ger 3
- ▶ Om en eller båda operanderna är ett flyttal, resultatet blir också ett flyttal; t.ex. 7.5/3 ger 2.5
- ▶ Om båda operanderna i en division är heltal, blir resultatet en heltalsdivision, dvs, vi göra oss av med fraktionsdel 10/3 ger 3



Prioritet

Prioritet	Operator	associerar
1	unärt: +, -	$h \rightarrow v$
2	binärt: *, /, %	$v \rightarrow h$
3	binärt: +, -, sträng sammanbindning	$v \rightarrow h$

`r = 77 - 33 * 2`

- Använd parenteser för att forcera ordningen !



Outline



Inkrementera och kombinera tilldelning med en operator

Flera kombinationer är möjliga i Java:

```
result = 0;
result++;
result+= 15 - 9 / 3;
result*= 2;
result--;
System.out.println("result= " + result);
```



Inmatning med Scanner

- **Scanner** används vid inmatning från tangentbord eller en fil ...
- Bryta ner inputen i en sekvens av lexikala element eller "tokens"

```
/* Scan.java
 * Programmet demonstrerar användandet av Scanner programm
 */
import java.util.Scanner;
class Scan
{
    // Metoden bryter ner en sträng i flera tokens och skriva ut de
    static public void main(String[] args){
        String source="1 TDDC 54 true 4.5 ett två tre!";
        Scanner scan = new Scanner (source);

        String token1 = scan.next();
        String token2 = scan.next();
        int token3 = scan.nextInt();
        boolean token4 = scan.nextBoolean();
        double token5 = scan.nextDouble();
        String token6 = scan.nextLine();

        System.out.println("token1 = " + token1);
        System.out.println("token2 = " + token2);
        System.out.println("token3 = " + token3);
        System.out.println("token4 = " + token4);
        System.out.println("token5 = " + token5);
        System.out.println("token6 = " + token6);
    }
}
```



Inmatning med Scanner

- ▶ **System.in** representerar "standard input stream", här tangentbordet
- ▶ kan också läsa in från en fil (vi tittar på det senare i kursen)

```
/* ScanStandard.java
 * Programmet demonstrerar hur man kan läsa in
 * från den standard input, ofta konsolen
 */
import java.util.Scanner;
class ScanStandard{
    /* Metoden läser in avstånd och tid från användaren,
     * räknar upp hastigheten och skriva ut den
     */
    static public void main(String[] args){
        Scanner scan = new Scanner (System.in);

        System.out.println("What was the distance in kms?");
        Double distance = scan.nextDouble();

        System.out.println("What time in minutes did it take?");
        Double time = scan.nextDouble();

        System.out.println("The average speed was: " + (distance*60/time) + "km/h");
    }
}
```



Strängar

- ▶ Exempel på en icke primitiv datatyp, eller en så kallade referenstyper
- ▶ "Rader av tecken", "TDDC77:are"
- ▶ Man kan inte ändra på en sträng efter att ha skapat en (immutable, diskuteras senare i kursen)



Outline

Strängar

```
/* Str.java
 * Programmet demonstrerar hur man kan använda sig
 * av Java APIet för att manipulera strängar
 */
class Str{

    // Metoden anropar på olika Sträng metoder från APIet
    static public void main(String[] args){
        String kursnamn = "TDDC77", are = ":are!";

        System.out.println("char at index 1: " + kursnamn.charAt(1));

        System.out.println("compare to \"TDDC77\" gives: " + kursnamn.compareTo("TDDC77"));

        System.out.println("compare to \"Z\" gives: " + kursnamn.compareTo("Z"));

        System.out.println("Hej " + kursnamn.concat(are));
        System.out.println("Hej " + kursnamn + are);

        System.out.println("TDDC77 equals tddc77: " + kursnamn.equals("tddc77"));

        System.out.println("TDDC77 equals ignore case: "
            + kursnamn.equalsIgnoreCase("tddc77"));

        System.out.println("Length of TDDC77: " + kursnamn.length());

        System.out.println("Replace D by d in TDDC77: "
            + kursnamn.replace('D', 'd'));

        System.out.println("Substring of TDDC77 from 2 to 5: "
            + kursnamn.substring(2,5));

        System.out.println("To lower case: " + kursnamn.toLowerCase());

        System.out.println("To upper case: " + are.toUpperCase());
    }
}
```



Application Programming Interface (API)

- ▶ <http://docs.oracle.com/javase/8/docs/api/>
- ▶ här hittar ni beskrivningar av alla “program” som kommer med Java (klasser, deras konstruktörer, metoder, attribut och så vidare).
- ▶ de vanligast använda klasserna finns i paketen `java.lang`, `java.io`, `java.util` och `java.text`.
- ▶ lär er hitta här så fort som möjligt.

