

Nyheter i Java 5



2.0

Huvudbegrepp

- Kollektioner med typer (Generics)
- Förbättrade for loopen
- Autoboxing och Unboxing
- Typsäkra Enum-satser
- Statiska Import-satser
- Input m.h.a. Scanner



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacoek Malec

2

Kollektioner med typer (Generics)

- Kollektioner använder polymorfism för att lagra objekt av godtyckliga typer.
- Vid objektakcess förlorar man typen:
`Lot selected = (Lot)lots.get(i);`
- Att använda otypade kollektioner betraktas som 'osäkerhet'.
- Typade kollektioner undviker typförlust.
- Körtidstesterna är enklare eftersom typen är känd.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacoek Malec

3

Ny syntax

- Typen av objekt som skall lagras anges vid deklaration:
`private ArrayList<String> notes;`
- ... och vid skapande:
`notes =
 new ArrayList<String>();`
- Kollektion-typer är parametriserade.



Objects First with Java - A Practical Introduction using BlueJ, © David J. Barnes, Michael Kölling
Svenska versionen av Eric Astor och Jacoek Malec

4

Användning av en kollektion med en typ

- Eftersom kollektionen har en typ behövs det ingen kastning (typomvandling):

```
Lot selection = lots.get(i);
```



Iteration med en typ

- En iterator med typ kan fås utav en kollektion med typ:

```
• Iterator<Lot> it = lots.iterator();  
while(it.hasNext()) {  
    Lot lot = it.next();  
    System.out.println(lot);  
}
```



HashMappar med typer

- HashMappar använder par (nyckel, värde).
- En HashMap med typ kräver två typparametrar:

```
private HashMap<String, String> responses;  
...  
responses = new HashMap<String, String> ();
```



Förbättrad for-loopen

- Iterering över arrayer och kollektioner är en typisk operation.
- Det händer ofta att avslutningsvillkor blir felaktiga.
- En förbättrad for loop förenklar iterationen.
 - Endast om loopvariabeln behövs ej!



Förbättrad for-loopen

Ursprungligt

```
for(int i = 0; i < array.length; i++) {  
    Type element = array[i];  
    ...  
}
```

Förbättrad

```
for(Type element : array) {  
    ...  
}
```

'för varje element i arrayen'



For-loopen med en Iterator

Ursprungligt

```
Iterator it = entries.iterator();  
while(it.hasNext()) {  
    System.out.println(it.next());  
}
```

Förbättrad

```
for(LogEntry entry : entries) {  
    System.out.println(entry);  
}
```



Autoboxing och Unboxing

- Omslagsklasserna (Wrapper classes) behövs inte längre när man använder primitiva typer i objekt-kontext.

- Autoboxing:

```
int i = 18;  
myCollection.add(i);
```

- Unboxing:

```
int i = myCollection.get(0);
```



Typsäkra enumreringar

- En ny syntax som tillåter definition av enummerade typer.
- Normalt skall ersätta en lista av nämnda, relaterade heltal, som t.ex.:

```
public static final int  
    LOW = 0, MEDIUM = 1,  
    HIGH = 2;
```



En enumrerad typ

```
public enum Level
{
    LOW, MEDIUM, HIGH
}
```

- Värdena hanteras som statiska variabler, t.ex. `Level.LOW`
- Värdena är skilda från heltal (ints) och har inget associerat numeriskt värde.



Statiska Import-satser

- Tillåter importen av statiska metoder och variabler, t.ex.
`import static java.lang.Math.abs;`
`import static java.lang.Math.PI;`
- Importerade namn får användas utan kvalificering.
- En ganska liten utökning av språket.



Input m.h.a. en Scanner

- Definierad i `java.util` paketet.
- Tillåter inläsning och bearbetning av indata.
- Ersätter existerande klasskombinationer som t.ex. `BufferedReader` med `StringTokenizer` eller `String.split`.
- Parsar indata och returnerar värde med typer.



Ett exempel med Scanner

```
// Scan a line of text for ints.
Scanner tokenizer = new Scanner(line);
for(int i = 0; i < dataline.length; i++) {
    dataline[i] = tokenizer.nextInt();
}
```



Sammanfattning

- Ett antal nya språkkonstruktioner och klasser finns i Java 5.
- Kollektioner med typer (generics) är troligen den mest viktiga innovationen.
- Den förbättrade for-loopen är nyttig.
- Enumrerade typer förbättrar läsbarheten och typsäkerheten av koden.

