

Att gruppera objekt

Kollektioner och iteratorer



2.0

Nya begrepp

- Kollektioner (Collections)
- Repetitionssatser (Loops)
- Iteratorer (Iterators)
- Arrayer (Arrays)



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

2

Krav på att gruppera objekt

- Många applikationer innehåller samlingar av objekt:
 - Personliga almanackor.
 - Bibliotekskataloger.
 - Studentregistreringssystem (LADOK).
- Antal element att lagra ändras.
 - Element sätts in.
 - Element tas bort.



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

3

Ett personligt anteckningsblock

- Anteckningar kan lagras.
- Individuella anteckningar kan synas.
- Det finns ingen begränsning av antalet anteckningar.
- Man måste få veta hur många anteckningar finns.
- Undersök notebook1-projektet.



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

4

Klassbibliotek

- Samling av användbara klasser.
- Innebär att man inte måste skriva allt själv.
- I Java kallas de olika biblioteken paket (eng. packages).
- Man har ofta behov av att hantera objekt som skall grupperas på något sätt.
 - Paketet `java.util` innehåller klasser som underlättar detta.



```
import java.util.ArrayList;

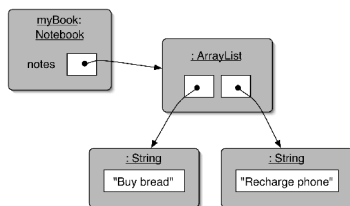
/**
 * ...
 */
public class Notebook
{
    // Storage for an arbitrary number of notes.
    private ArrayList notes;

    /**
     * Perform any initialization required for the
     * notebook.
     */
    public Notebook()
    {
        notes = new ArrayList();
    }

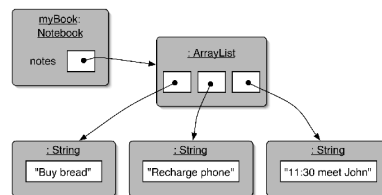
    ...
}
```



Kollektioner av objekt



Sätt in en tredje anteckning



Fördelar med en kollektion

- Kapaciteten ökar när det behövs.
- Håller reda på antalet objekt i kollektionen (`size()` accessor).
- Ser till att ordningen mellan objekten upprätthålls.
- Detaljerna hos implementeringen är dold
 - Är det bra? Om vi inte vet hur det är implementerat vågar vi använda det?
Ja - men vi måste känna till specifikationen av det vi använder.



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

9

Användning av kollektioner

```
public class Notebook
{
    private ArrayList notes;
    ...

    public void storeNote(String note)
    {
        notes.add(note); // Insättning av en anteckning
    }

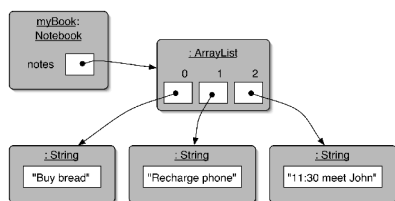
    public int numberOfNotes()
    {
        return notes.size(); // Returnerar antalet anteckningar (delegation).
    }
    ...
}
```



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

10

Indexerad numrering



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

11

Att komma åt ett objekt

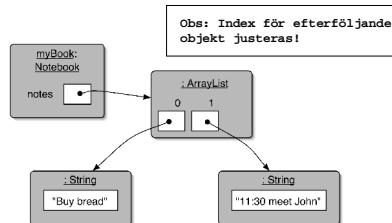
```
public void showNote(int noteNumber)
{
    // Index validity checks
    if(noteNumber < 0) {
        // This is not a valid note number.
    }
    else if(noteNumber < numberOfNotes()) {
        System.out.println(notes.get(noteNumber));
    }
    else {
        // This is not a valid note number.
    }
}
// Retrieve and print the note
```



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

12

Borttagning av ett objekt



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

13

remove(<index>)

```
/**
 * Removes the element at the specified position in this list.
 * Shifts any subsequent elements to the left (subtracts one from their
 * indices).
 *
 * @param index the index of the element to be removed.
 * @return the element that was removed from the list.
 * @throws IndexOutOfBoundsException if index out of range <tt>(index
 *         < 0 || index >= size())</tt>.
 */
public Object remove(int index) {
    RangeCheck(index);

    modCount++;
    Object oldValue = elementData[index];
    int numMoved = size - index - 1;
    if (numMoved > 0)
        System.arraycopy(elementData, index+1, elementData, index,
            numMoved);
    elementData[--size] = null; // Let gc do its work

    return oldValue;
}
```

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

14

while-exempel

```
/**
 * List all notes in the notebook.
 */
public void listNotes()
{
    int index = 0;
    while(index < notes.size()) {
        System.out.println(notes.get(index));
        index++;
    }
}
```

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

15

Kollektioner

- Kollektioner tillåter ett godtyckligt antal objekt att lagras i en struktur.
- Klassbibliotek innehåller väl uttestade klasser. Risken för fel är mycket liten.
- Javas klassbibliotek kallas packages.
- Vi har tittat på `ArrayList` klassen från `java.util` paketet.
- Man itererar bäst över kollektioner med hjälp av en iterator.

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

16

Oversikt

- Element kan sättas in och tas bort.
- Varje element har en index.
- Indexvärde kan ändras om element tas bort (eller om nya element sätts in).
- Huvudmetoder i `ArrayList` är `add`, `get`, `remove` och `size`.

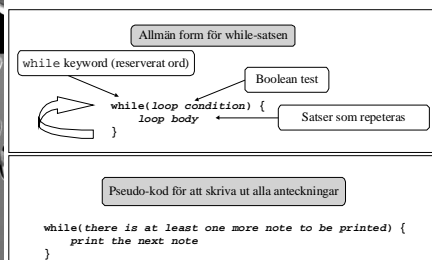


Iteration

- Ofta vill vi utföra vissa aktioner ett godtyckligt antal gånger.
 - T.ex. Skriv ut alla anteckningar i ett anteckningsblock. Hur många finns det?
- De flesta programmeringsspråk innehåller loop-satser för att kunna göra det.
- Java har tre sorters loop-satser.
 - Vi skall fokusera på while-satsen.



While-satsen



Ett exempel

```
/**  
 * List all notes in the notebook.  
 */  
public void listNotes()  
{  
    int index = 0;  
    while (index < notes.size()) {  
        System.out.println(notes.get(index));  
        index++;  
    }  
}
```

Diagram label: Increment by one (pointing to `index++`)



Iterator-metoder

Metoder som förutsätts ingå i ett iterator-objekt:

```
boolean hasNext()  
    Returns true if the iteration has more elements.  
  
Object next()  
    Returns the next element in the iteration.  
  
void remove()  
    Removes from the underlying collection the last  
    element returned by the iterator (optional operation).
```



Iterering över en kollektion

```
java.util.Iterator  
    Returnerar ett Iterator-objekt  
  
Iterator it = myCollection.iterator();  
while(it.hasNext()) {  
    call it.next() to get the next object  
    do something with that object  
}
```

```
public void listNotes()  
{  
    Iterator it = notes.iterator();  
    while(it.hasNext()) {  
        System.out.println(it.next());  
    }  
}
```



auction-projektet

- auction-projektet ger en utökad bild av kollektioner och iteratorer.
- Två saker att tänka på:
 - null-värdet.
 - Casting. Används för att lagra resultat av get i en variabel:
 - `String message = (String) notes.get(0);`



Överblick

- Loop-satser tillåter ett block att upprepas.
- I en while-sats styrs upprepningen av ett booleskt uttryck.
- Kollektion-klasser har speciella Iterator-objekt som underlättar itereringen över hela kollektionen.



Kollektioner med en bestämd storlek

- Ibland kan den maximala storleken på en kollektion bestämmas i förväg.
- Programmeringsspråk brukar erbjuda en speciell kollektion med fastställd storlek: arrayer.
- Javas arrayer kan lagra objekt eller värde av primitiva typer.
- Arrayer använder en särskild syntax.



weblog-analyzer-projektet

- Webbservern sparar detaljer av varje access.
- Webbmasters uppgifter:
 - Mest populära sidor.
 - Rulsperioder.
 - Hur mycket data blir levererad.
 - Brutna länkar.
- Analysera åtkomster för varje timme.



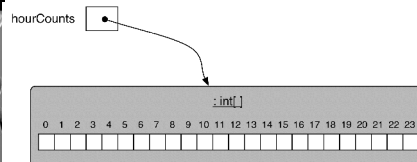
Att skapa en array

```
public class LogAnalyzer
{
    private int[] hourCounts; ← Array variable declaration
    private LogfileReader reader;

    public LogAnalyzer()
    {
        hourCounts = new int[24]; ← Array object creation
        reader = new LogfileReader();
    }
    ...
}
```



The hourCounts array



array

- Kantiga parenteser används för att referera elementen: `hourCounts[...]`
- Elementen används på samma sätt som vanliga variabler.
 - Till vänster i en tilldelning:
`.hourCounts[hour] = ...;`
 - I ett uttryck:
`.adjusted = hourCounts[hour] - 3;`
`.hourCounts[hour]++;`



for-satsen

- Fungerar som en specialiserad while-sats.
- Används när antalet repetitioner är känt i förväg.
- Mycket lämplig när man behandlar elementen i en array.



for-satsen

Allmän form

```
for(initialization; condition; post-body action) {  
    statements to be repeated  
}
```

Motsvarande while-sats

```
initialization;  
while(condition) {  
    statements to be repeated  
    post-body action  
}
```



Ett exempel

Exempel med for-sats

```
for(int hour = 0; hour < hourCounts.length; hour++) {  
    System.out.println(hour + ": " + hourCounts[hour]);  
}
```

Samma exempel med while-sats

```
int hour = 0;  
while(hour < hourCounts.length) {  
    System.out.println(hour + ": " + hourCounts[hour]);  
    hour++;  
}
```





Sammanfattning

- arrayer används med fördel när en kollektion har ett fixt antal element,
- arrayer har en speciell syntax,
- for-satser är bra och tydliga att använda när antalet repetitioner är känt i förväg,
- for-satser är speciellt lämpliga när man vill behandla element i en array.

