

## Processing a Collection

### 1 Using a Loop

From C we know two types of loops:

-while  
-for

These loops also apply in Java and can be used to process a collection. But Java offers many possibilities to loop through a list of elements.

### 2 For-each Loop

```
for (Dog oneDog : myDogs2){  
    oneDog.bark();  
}
```

This basically reads “for each dog in myDogs, do the following”.

It is a very elegant way to process a collection, if you are not addressing a specific element in the collection.

→ **This is the standard technique, if all elements of a collection should be processed.**

### 3 For Loop

This is the “classic” way of dealing with a for-loop. We have seen this style in C as well. It is used for a primitive array:

```
for (x=0; x < myDogs.length; x++){  
    myDogs[x].bark();  
    myDogs[x].tellAge();  
}
```

Or for a collection (here an ArrayList):

```
for (x=0; x < myDogs2.size(); x++){  
    myDogs2.get(x).bark();  
    myDogs2.get(x).tellAge();  
}
```

Note how we have to use the methods from a collection to use the index.

→ **Using an index does not necessarily work with all collection classes!**

## 4 While Loop

```
int index = 0;
while(index < myDogs2.size()){
    Dog d = myDogs2.get(index);
    d.bark();
    index++;
}
```

The while-loop can do the same as the for-each loop, but you are working here with an index. And you have to do more yourself:

- a) You have to assign each element from the list to a variable
- b) You have to increment the index (counter).

This loop is more useful if you only want to process a part of a collection.

→ **But you cannot always use this method for all collections.**

For some collections it is either impossible or very inefficient to access individual elements by using an index.

## 5 While Loop Using an Iterator

```
//or with an iterator:
Iterator<Dog> iterate = myDogs2.iterator();
while (iterate.hasNext()){
    Dog doggy = iterate.next();
    doggy.bark();
}
```

This solution with an Iterator is available for all collections and is an important code pattern (“Programmier-Muster”).