Java for loop syntax: "for (T obj : objects)"

Asked 9 years, 7 months ago   Active 4 years, 11 months ago   Viewed 151k times

▲

50

▼

I came across some Java syntax that I haven't seen before. I was wondering if someone could tell me what's going on here.

```
for (ObjectType objectName : collectionName.getObjects())
```

🔖

22

🕘

java   syntax   for-loop   foreach

Share Edit Follow

7 Answers

Active | Oldest | Votes

▲

98

▼

It's called a **for-each** or **enhanced** `for` **statement.** See the JLS §14.14.2.

It's **syntactic sugar** provided by the compiler for iterating over `Iterable`s and arrays. The following are equivalent ways to iterate over a list:

✓

🕘

```
List<Foo> foos = ...;
for (Foo foo : foos)
{
    foo.bar();
}

// equivalent to:
List<Foo> foos = ...;
for (Iterator<Foo> iter = foos.iterator(); iter.hasNext();)
{
    Foo foo = iter.next();
    foo.bar();
}
```

and these are two equivalent ways to iterate over an array:

```
int[] nums = ...;
for (int num : nums)
{
    System.out.println(num);
}

// equivalent to:
int[] nums = ...;
for (int i=0; i<nums.length; i++)
{
    int num = nums[i];
    System.out.println(num);
}
```

Further reading

Share Edit Follow

5

The variable **objectSummary** holds the current object of type **S3ObjectSummary** returned from the **objectListing.getObjectSummaries()** and iterate over the collection.

Here is an example of this *enhanced for loop* from [Java Tutorials](#)

```java
class EnhancedForDemo {
 public static void main(String[] args){
      int[] numbers = {1,2,3,4,5,6,7,8,9,10};
      for (int item : numbers) {
        System.out.println("Count is: " + item);
      }
 }
}
```

In this example, the variable item holds the current value from the numbers array.

Output is as follows:

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10
```

Hope this helps !

Share Edit Follow

4

yes... This is for each loop in java.

Generally this loop is become useful when you are retrieving data or object from the database.

Syntex :

```java
for(Object obj : Collection obj)
{
     //Code enter code here
}
```

Example :

```
for(User user : userList)
{
    System.out.println("USer NAme :" + user.name);
  // etc etc
}
```

This is for each loop.

it will incremental by automatically. one by one from collection to USer object data has been filled. and working.

Share Edit Follow

2

That's the for each loop syntax. It is looping through each object in the collection returned by
`objectListing.getObjectSummaries()` .

Share Edit Follow

2

for each S3ObjecrSummary in objectListing.getObjectSummaries()

it's looping through each item in the collection

Share Edit Follow

1

```
public class ForEachLoopExample {

    public static void main(String[] args) {

        System.out.println("For Each Loop Example: ");

        int[] intArray = { 1,2,3,4,5 };

        //Here iteration starts from index 0 to last index
        for(int i : intArray)
            System.out.println(i);
    }

}
```

Share Edit Follow

Excessive promotion of a specific product/resource may be perceived by the community as spam. Take a look at the help center, specially What kind of behavior is expected of users?'s last section: Avoid overt self-promotion. You might also be interested in How do I advertise on Stack Overflow? – Draken May 26 '16 at 7:48 ✏

1

Yes, It is called the for-each loop. Objects in the collectionName will be assigned one after one from the beginning of that collection, to the created object reference, 'objectName'. So in each iteration of the loop, the 'objectName' will be assigned an object from the 'collectionName' collection. The loop will terminate once when all the items(objects) of the 'collectionName' Collection have finished been assigning or simply the objects to get are over.

for (ObjectType objectName : collectionName.getObjects()){ //loop body> //You can use the 'objectName' here as needed and different objects will be //reepresented by it in each iteration. }

Share Edit Follow

answered Dec 18 '13 at 14:16

Samitha Chathuranga
1,536   4   25   56