



# Percorrendo coleções

---

- É possível usar um loop baseado em índice para percorrer uma coleção?

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        for(int i = 0; i < lista.size(); i++){
            String estado = lista.get(i);
            System.out.println(estado);
        }
    }
}
```

O código acima funciona?

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        LinkedList<String> lista = new LinkedList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        for(int i = 0; i < lista.size(); i++){
            String estado = lista.get(i);
            System.out.println(estado);
        }
    }
}
```

O código acima funciona?



# Percorrendo coleções

---

- Se é possível utilizar um loop baseado em índices, por que usar um outro esquema de iteração?
  - Algumas coleções não armazenam os elementos em “ordem” (quem é o 1º, 2º, 3º, etc).

Quais coleções?

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        HashSet<String> lista = new HashSet<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        for(int i = 0; i < lista.size(); i++){
            String estado = lista.get(i);
            System.out.println(estado);
        }
    }
}
```

O código acima funciona?

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        TreeSet<String> lista = new TreeSet<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        for(int i = 0; i < lista.size(); i++){
            String estado = lista.get(i);
            System.out.println(estado);
        }
    }
}
```

O código acima funciona?



# Iterador em Java

---

- O que é um iterador em Java?
  - É a forma usada para percorrer os elementos de uma coleção.
- Como percorrer uma coleção em Java?
  - **while**
  - for
  - for-each

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i = lista.iterator();
        ➡ while(i.hasNext()) {

            }
        }
    }
}
```

hasNext() **NÃO** avança para o próximo elemento

`hasNext()`

Returns true if the iteration has more elements.



```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i = lista.iterator();
        while(i.hasNext()) {
            ➡ String estado = i.next();


        }
    }
}
```

next() avança para o próximo elemento

`next()`

Returns the next element in the iteration.

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i = lista.iterator();
        while(i.hasNext()) {
            String estado = i.next();
             System.out.println(estado);
        }
    }
}
```

Imprimindo o estado

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i = lista.iterator();
        while(i.hasNext()) {
            ➡ System.out.println(i.next());
        }
    }
}
```

Outra forma de impressão direta



# Iterador em Java

---

- O que é um iterador em Java?
  - É a forma usada para percorrer os elementos de uma coleção.
- Como percorrer uma coleção em Java?
  - while
  - **for**
  - for-each

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i = lista.iterator();
        while(i.hasNext()) {
            System.out.println(i.next());
        }
    }
}
```

Como transformar o laço WHILE em um laço FOR tradicional?

```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");

        Iterator<String> i;
        ➡ for(i = lista.iterator(); i.hasNext();){
            System.out.println(i.next());
        }
    }
}
```



# Iterador em Java

---

- O que é um iterador em Java?
  - É a forma usada para percorrer os elementos de uma coleção.
- Como percorrer uma coleção em Java?
  - while
  - for
  - **for-each**



# enhanced-for

---

- O enhanced-for é uma versão simplificada do laço for, também usada para percorrer coleções.
- Sintaxe

```
for(<Tipo> <identificador> : <array ou coleção>)  
{  
    <comando>  
}
```



```
import java.util.*;
public class Principal{
    public static void main(String[] args)
    {
        ArrayList<String> lista = new ArrayList<String>();
        lista.add("São Paulo");
        lista.add("Paraná");
        lista.add("Santa Catarina");
        lista.add("São Paulo");
        ➡ for(String cidade : lista)
        {
            System.out.println(cidade);
        }
    }
}
```

```
public class Principal{  
    public static void main(String[] args) {  
        Empregado emp1 = new Empregado("Jose",130);  
        Empregado emp2 = new Empregado("Ana", 110);  
        Empregado emp3 = new Empregado("Jose", 130);  
  
        Collection<Empregado> c = new TreeSet<Empregado>();  
        c.add(emp1);  
        c.add(emp2);  
        c.add(emp3);  
  
        Iterator<Empregado> i = c.iterator();  
        while ( i.hasNext() )  
        {  
            Empregado e = i.next();  
            S.O.P(e.getNome() + " " + e.getSalario());  
        }  
    }  
}
```

Percorra a coleção usando o FOR-EACH

```
public class Principal{  
    public static void main(String[] args) {  
        Empregado emp1 = new Empregado("Jose",130);  
        Empregado emp2 = new Empregado("Ana", 110);  
        Empregado emp3 = new Empregado("Jose", 130);  
  
        Collection<Empregado> c = new TreeSet<Empregado>();  
        c.add(emp1);  
        c.add(emp2);  
        c.add(emp3);  
  
        → for( Empregado e :c )  
        {  
            S.O.P(e.getNome() + " " + e.getSalario());  
        }  
    }  
}
```