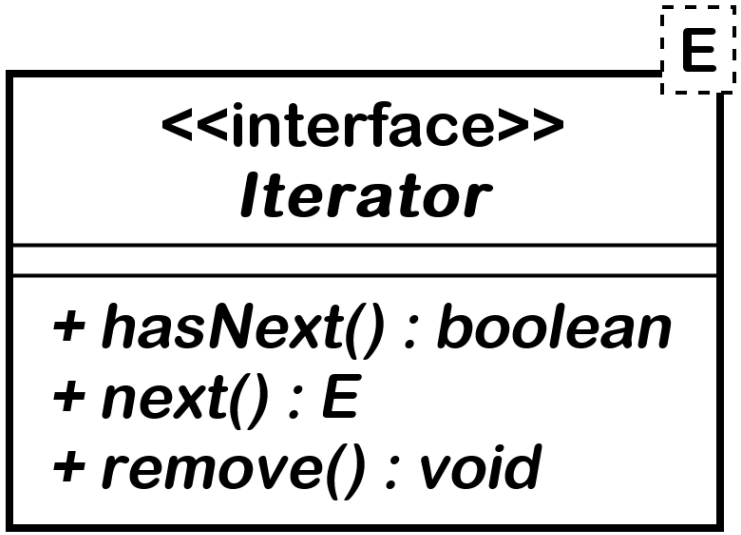
O método iterator()

- A interface **`Collection`** prevê que todas as classes concretas que a implemente, retorne um *iterador*, através do método **`iterator()`**.



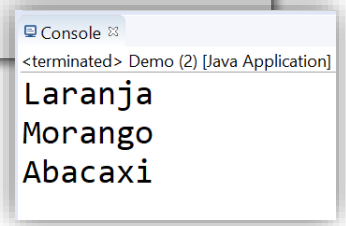
Um *iterador* é um objeto que possibilita percorrer uma estrutura de dados.

Método	Descrição
hasNext()	Retorna true se tem objetos na coleção ainda não lidos
next()	Retorna um objeto da coleção
remove()	Remove o último objeto lido pelo iterador (através de next()), da coleção

Utilidade do iterador

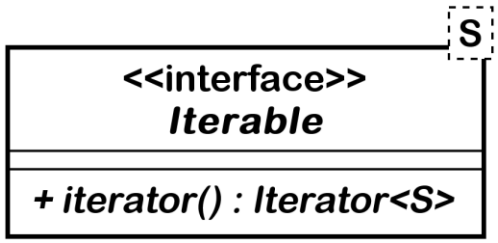
- Usando o iterador:

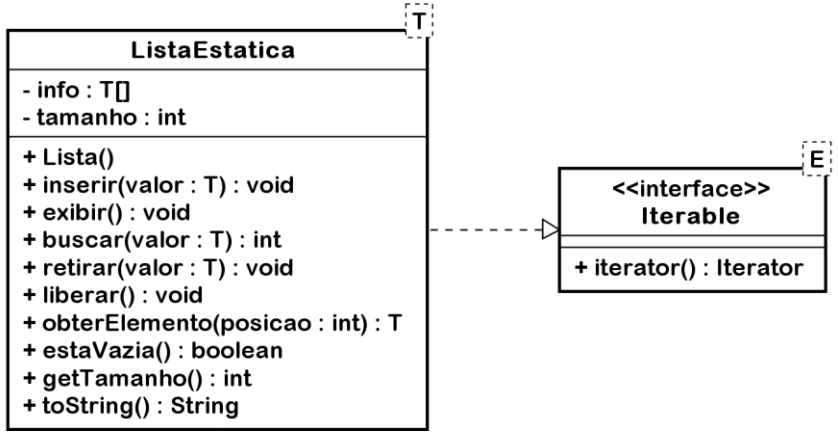
```
Collection<String> frutas = new ArrayList<>();
frutas.add("Laranja");
frutas.add("Morango");
frutas.add("Pêssego");
frutas.add("Pêra");
frutas.add("Abacaxi");

Iterator<String> iterador = frutas.iterator();
while (iterador.hasNext()) {
    String fruta = iterador.next();
    if (fruta.startsWith("P"))
        iterador.remove();
}

for (String fruta : frutas) {
    System.out.println(fruta);
}
```



Utilidade do iterador

Qualquer classe que implementa a interface **`Iterable`** está apta a ser utilizada num *for each*:






```
ListaEstatica<String> nomes;
nomes = new ListaEstatica<>();
nomes.inserir("Joao");
nomes.inserir("Maria");

for (String nome : nomes) {
    System.out.println(nome);
}
```

Utilidade do iterador