

Mãos na massa: Vector e Collection

Vamos praticar rapidamente o que vimos nessa aula mais conceitual. Para isso, siga os passos abaixo.

1) Abra a classe `TesteArrayList` para testar as outras implementações de listas. Para tal, instancie um `java.util.Vector`, mas também teste `ArrayList` e `LinkedList`:

```
public class TesteArrayList {  
  
    public static void main(String[] args) {  
  
        //List<Conta> lista = new ArrayList<Conta>(); //usa array por baixo  
        //List<Conta> lista = new LinkedList<Conta>(); //lista linkada  
        List<Conta> lista = new Vector<Conta>(); //usa array por baixo, mas tbm é threadsafe  
  
        //outro código omitido  
        //...  
    }  
}
```

Repare que todo o resto do código continua funcionando, independente da lista utilizada. Isto ocorre pois todas as implementações são listas.

2) Teste uma vez a interface `java.util.Collection` na mesma classe `TesteArrayList`:

```
public class TesteArrayList {  
  
    public static void main(String[] args) {  
  
        //List<Conta> lista = new ArrayList<Conta>(); //usa array por baixo  
        //List<Conta> lista = new LinkedList<Conta>(); //lista linkada  
        Collection<Conta> lista = new Vector<Conta>();  
  
        //outro código omitido  
        //...  
        Conta ref = lista.get(0); //não compila  
        //...  
    }  
}
```

Repare que algumas linhas de código param de compilar, justamente os métodos que usam a posição, como `remove(int pos)` ou `get(int pos)`. Isto ocorre pois apenas listas são sequências e possuem um índice, mas há outras coleções que funcionam sem índice e consequentemente não possuem esses métodos.

3) Após testar a interface `java.util.Collection` volte a usar a interface `java.util.List`.

