**COLLECTIONS FRAMEWORK**

**Introduction**

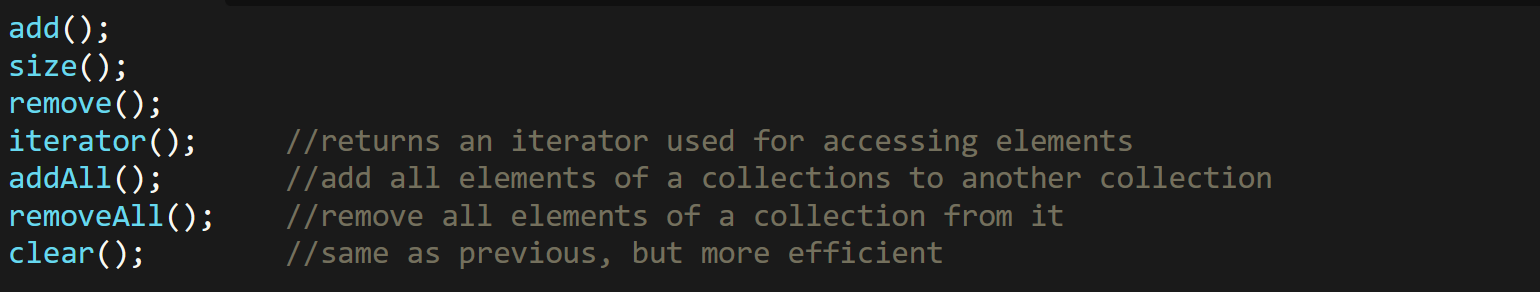
* Used to implement various **DSAs**.
* ***java.util*** contains all **collections** **classes** and **interfaces**.

**Collections Framework Hierarchy**



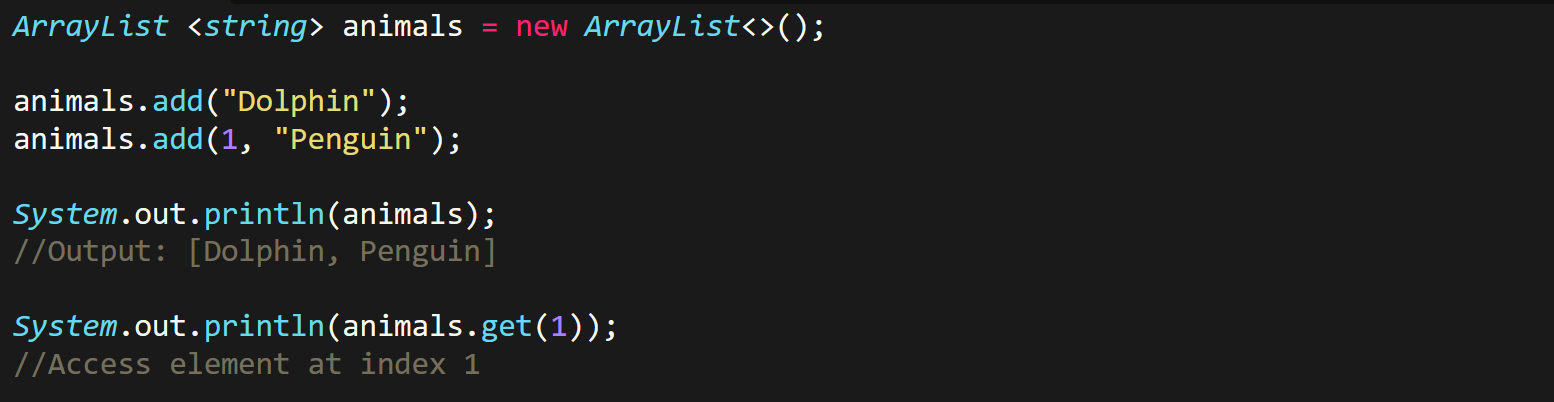
* The **blue** ones (interfaces) are also known as **collection interface**.
* And the **greens** (classes) are also known as **collection subinterfaces**.

**Common Collection Methods**



**ArrayList**

* Provides facility for **resizable** array (also in vectors).



* Never write ***int*** i.e. data type inside the angle brackets, write ***Integer*** i.e. **wrapper**.

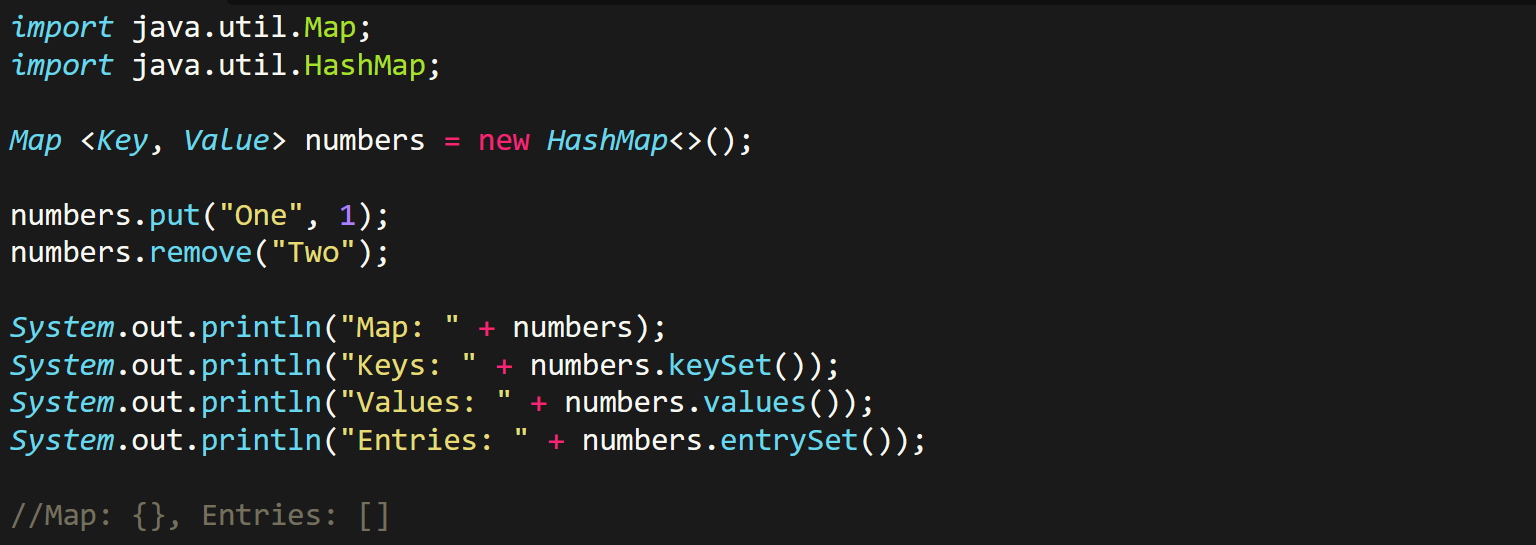
**ArrayList VS Vector**

* **Synchronized:** Sharable among threads.
* ArrayList uses **non-synchronized** methods.
* Vectors use **synchronized** methods.
* ArrayList **never** loses efficiency as elements increase.
* Vector **loses** efficiency as elements increase.
* ArrayList faces **no** thread access collision.
* Vector **faces** thread access collision.

**Maps**

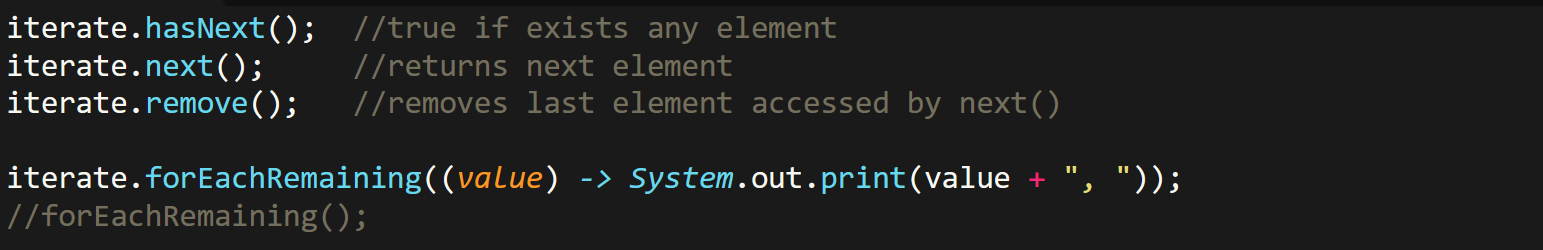
* **Containing sets:**
  + Keys
  + Values
  + Mapping





**Iterator**

* Contains **sub-interface** called ***ListIterator***.



* An iterator is created, and then assigned a collection for iterating:

