

کتابخانه های Guava و Apache commons  
در کتابخانه Guava :

Collection , Caching , String , Primitive , Math  
پکیج ها موجود است .

Guava مجموعه های پیشرفته بسیاری را بر اساس تجربه توسعه دهندگان در کارهای توسعه برنامه ها معرفی می کند.

در زیر لیستی از مجموعه های مفید collection ارائه شده است:

منبع جدول ها : <https://www.tutorialspoint.com/>

Sr.No	Collection name & Description
1	<u>Multiset</u> An extension to Set interface to allow duplicate elements.
2	<u>Multimap</u> An extension to Map interface so that its keys can be mapped to multiple values at a time.
3	<u>BiMap</u> An extension to Map interface to support inverse operations.
4	<u>Table</u> Table represents a special map where two keys can be specified in combined fashion to refer to a single value.

Guava مکانیزم ذخیره سازی مبتنی بر حافظه بسیار قدرتمند رابط LoadingCache <V, K> فراهم می کند. مقادیر

بطور خودکار در حافظه پنهان بار گذاری می شوند و Caching بسیاری از متد های کاربردی برای نیازهای ذخیره

سازی مفید فراهم می کند:

Sr.No	Method & Description
1	<b>V apply(K key)</b> Deprecated. Provided to satisfy the Function interface; use get(K) or getUnchecked(K) instead.

2	<b>ConcurrentMap&lt;K,V&gt; asMap()</b> Returns a view of the entries stored in this cache as a thread-safe map.
3	<b>V get(K key)</b> Returns the value associated with key in this cache, first loading that value if necessary.
4	<b>ImmutableMap&lt;K,V&gt; getAll(Iterable&lt;? extends K&gt; keys)</b> Returns a map of the values associated with keys, creating or retrieving those values if necessary.
5	<b>V getUnchecked(K key)</b> Returns the value associated with key in this cache, first loading that value if necessary.
6	<b>void refresh(K key)</b> Loads a new value for key, possibly asynchronously.

Guava بسیاری از ابزارهای پیشرفته String را بر اساس تجربه توسعه دهندگان در کارهای توسعه برنامه های کاربردی معرفی می کند. در زیر لیست برنامه های کاربردی مبتنی بر String آمده است :

Sr.No	Utility name & Description
1	<u>Joiner</u> Utility to join objects, string etc.
2	<u>Splitter</u> Utility to split string.
3	<u>CharMatcher</u> Utility for character operations.
4	<u>CaseFormat</u> Utility for changing string formats.

از آنجایی که از primitive type جاوا برای pass generic یا collections به عنوان ورودی استفاده نمی شود ،  
 Guava بسیاری از کلاس های Utility Wrapper را برای مدیریت کردن primitive type به عنوان Objects  
 ارائه می داد. در زیر لیستی از ابزارهای مفید primitive type مفید وجود دارد :

Sr.No	Utility name & Description
1	<u>Bytes</u> Utility for primitive byte.
2	<u>Shorts</u> Utility for primitive short.
3	<u>Ints</u> Utility for primitive int.
4	<u>Longs</u> Utility for primitive long.
5	<u>Floats</u> Utility for primitive float.
6	<u>Doubles</u> Utility for primitive double.
7	<u>Chars</u> Utility for primitive char.
8	<u>Booleans</u> Utility for primitive boolean.

Guava کلاسهای برنامه های مرتبط با ریاضیات را برای اداره int ، long و Big Integer فراهم می کند. در زیر  
 لیست آمده است:

Sr.No	Utility name & Description
1	<u>IntMath</u> Math utility for int.
2	<u>LongMath</u> Math utility for long.
3	<u>BigIntegerMath</u> Math utility for BigInteger.

در کتابخانه Apache commons

ویژگی های مختلفی را برای آسان تر کردن کار مجموعه فراهم می کند. `implementations` ، `interfaces` و `utilities` بسیاری را ارائه می دهد. ویژگی های اصلی مجموعه های زیر به شرح زیر است :

### Bag – Bag interfaces

`Collection` هایی که چندین کپی از `object` ها دارند را ساده تر می کند . رابط های جدید به کیسه های پشتیبانی اضافه می شوند. یک کیف مجموعه ای را تعریف می کند که تعداد دفعات نمایش یک شی را در مجموعه شمارش می کند. به عنوان مثال ، اگر یک کیسه حاوی `{a, a, b, c}` است ، می توانید `2` , `getCount("a")` را برگردانید ، در حالی که `uniqueSet()` مقادیر منحصر به فرد را برمی گرداند.

### BidiMap

رابط های `BidiMap` نقشه های دو جهته را ارائه می دهند که می توانند برای جستجوی مقادیر با استفاده از کلیدها یا کلیدها با استفاده از مقادیر استفاده شوند. رابط های جدید برای پشتیبانی از نقشه دو طرفه اضافه شده است. با استفاده از نقشه دو طرفه ، می توان با استفاده از کلید و با استفاده از کلید به راحتی کلید را جستجو کرد.

### MapIterator

رابط `MapIterator` تکرار ساده و آسان را بر روی نقشه ها ارائه می دهد. تکرار واسط `JDK Map` بسیار دشوار است زیرا `Iteration` در `EntrySet` یا اشیاء `KeySet` باید انجام شود. `MapIterator` تکرار ساده ای را از طریق `Map` فراهم می کند.

## Ordered Map

نقشه های سفارش یافته ترتیب عناصر اضافه شده را حفظ می کنند. `OrderedMap` یک رابط جدید برای نقشه ها برای حفظ ترتیب اضافه شدن عناصر است. `LinkedMap` و `ListOrderedMap` دو پیاده سازی موجود هستند. این رابط ها از تکرار نقشه استفاده می کنند و اجازه می دهد تا از هر دو جهت به جلو یا عقب در یک نقشه تکرار شود.

## Ordered Set

`Sets Ordered` ترتیب اضافه شدن عناصر را حفظ می کند.

## Comparator implementations

بسیاری از پیاده سازی های `Iterator` در دسترس هستند.

---

## String

با مطالعه ی مستندات برای متدهای :

`trim` یک `string` برمی گرداند که تمام `whitespace` های ابتدا و انتها حذف شده است.

`substring` دو نوع دارد یا فقط `index` ابتدایی را می گیرد و از آنجا به بعد را برمی گرداند یا `index` ابتدایی و پایانی را میگیرد و آن رشته را برمیگرداند .

`split` رشته را در اطراف عبارت داده شده تقسیم می کند ، برش می زند .

`replaceAll` جایگزینی تمام رشته های برابر با رشته ورودی با عبارت جایگزین .

`compareTo` دو رشته را با هم مقایسه می کند .

`toLowerCase` تمام کاراکتر های این رشته به حرف کوچک تبدیل می شوند .

`toUpperCase` تمام کاراکتر های این رشته به حرف بزرگ تبدیل می شوند .

`concat` دو رشته را به هم متصل می کند.

---

## تغییرات حافظه در visualVM



