

Core Java

آموزش ساده و آسان جاوا

به نام خدا

تقدیرم به هموطنان عزیزم

جاوا را با لذت یاد بگیرید!

Core Java

آموزش ساده و آسان جاوا

آموزش زبان برنامه نویسی جاوا

در جاوا Collections

موضوع: فریم ورک Collection

جلسه: اول

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!



این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.

Core Java

آموزش ساده و آسان جاوا

Collections in Java

سلام. دوست عزیز، از این که ما رو تا اینجا همراهی کردید صمیمانه سپاسگزارم. در این جلسه آموزشی قصد داریم با هم در مورد Collection ها در جاوا بحث کنیم. در فارسی به مجموعه یا کلکسیون هم می گویند.

Collection ها در جاوا چارچوبی (framework) است که معماری را برای برای ذخیره و دستکاری گروهی از اشیا فراهم کرده است.

همه عملیاتی که شما بر روی داده ها مثل جست و جو ، مرتب سازی ، درج کردن ، دستکاری ، حذف کردن و غیره انجام می دهید، می تواند توسط Collection ها در جاوا انجام شود.

Collection ها به معنای ساده واحدی از اشیا هستند که برای ذخیره سازی و دستکاری داده ها استفاده می شوند و به سادگی می توانیم عملیات های مذکور رو روی آنها انجام دهیم و داده های آنها را دستکاری کنیم.

فریمورک (framework) Collection در جاوا از تعدادی اینترفیس ها (interfaces) و کلاس ها (classes) تشکیل شده است.

اینترفیس های Collection شامل موارد زیر می شود:

Set, List, Queue, Deque,...

کلاس های Collection نیز شامل موارد زیر می شود:

ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet ,...

Core Java

آموزش ساده و آسان جاوا

Collection در جاوا چیست؟ Collection در جاوا نشان دهنده گروهی از اشیا می باشد. به عبارت دیگر برای نگهداری ، دستکاری و نمایش گروهی از اشیا استفاده می شود.

فریم ورک (framework) در جاوا چیست؟ یک فریم ورک جاوا شامل مجموعه ای از کتابخانه ها ، کلاس ها و اینترفیس ها می باشد.

Collection framework در جاوا چیست؟ Collection framework نشان دهنده یک معماری یکپارچه برای ذخیره و دستکاری گروهی از اشیا می باشد، که شامل موارد زیر است:

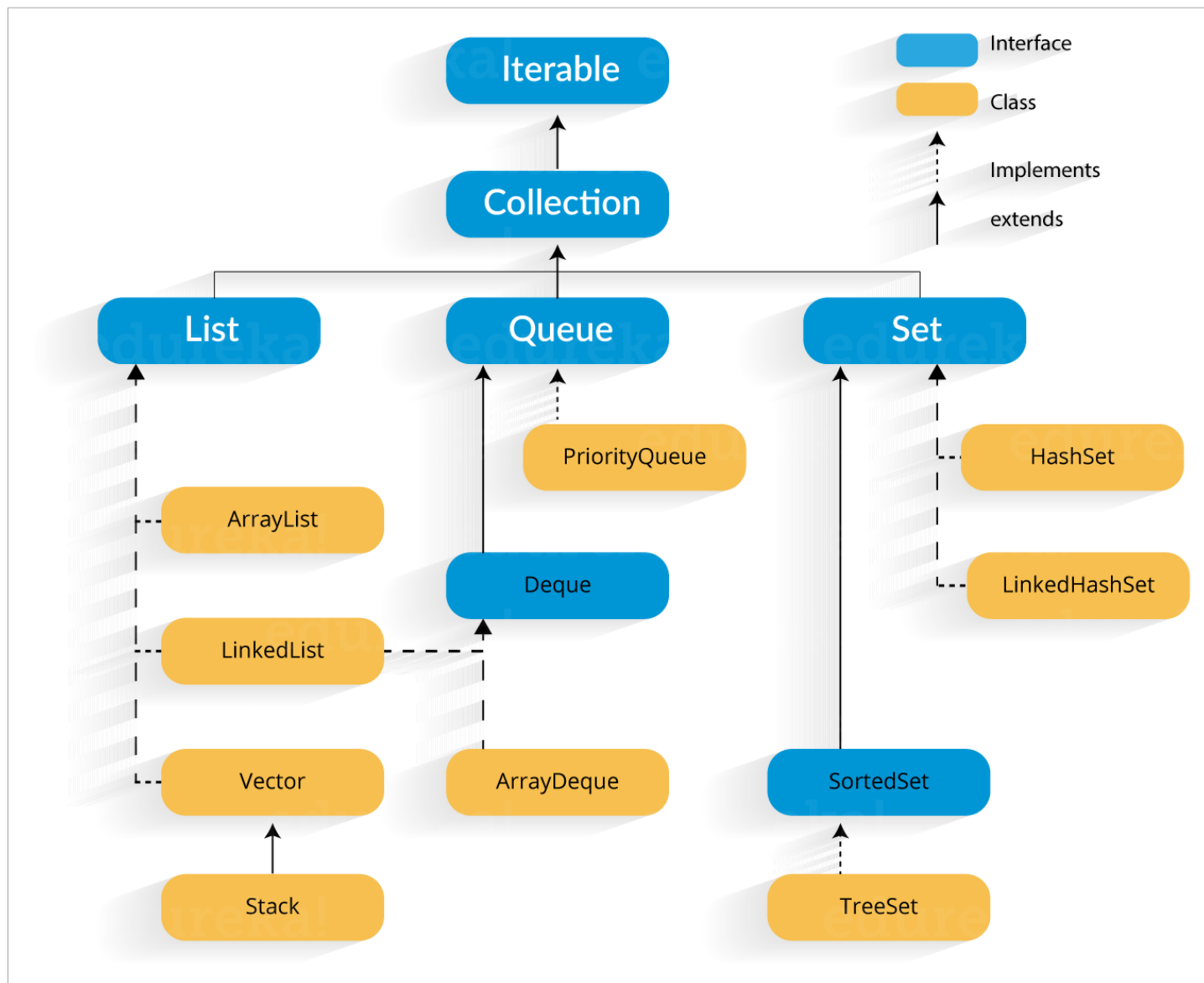
۱. اینترفیس ها (Interfaces) و کلاس هایی که به این اینترفیس ها implements شده اند یا به عبارتی دیگر کلاس هایی که این اینترفیس ها در آنها پیاده سازی شده است.
۲. الگوریتم (Algorithm) : الگوریتم به متدهای مفید و کاربردی که عملیات مرتب سازی ، جست و جو و... را روی اشیا موجود در Collection ها انجام می دهند اشاره دارد.

سلسله مراتب فریم ورک Collection در جاوا

سلسله مراتب Collection در جاوا را در تصویر (۱) جمع آوری کرده ایم. همان طور که در تصویر (۱) مشاهده می کنید، فریم ورک Collection مجموعه ای از اینترفیس ها و کلاس هایی که این اینترفیس ها را implements کرده اند و همچنین کلاس هایی که برخی کلاس ها را به ارث برده یا extends کرده اند را شامل می شود. ما جلسه به جلسه اینترفیس ها و کلاس های موجود در فریم ورک Collection را با قدرت بررسی خواهیم کرد.

Core Java

آموزش ساده و آسان جاوا



تصویر (۱)

متدهای موجود در اینترفیس Collection :

متدهای زیادی در اینترفیس Collection وجود دارد که آنها را در زیر جمع آوری کرده ایم:

Core Java

آموزش ساده و آسان جاوا

توضیح	متد	شماره
برای درج یک عنصر در collection استفاده می شود.	<code>public boolean add(Object element)</code>	۱
اگر یک collection این متد را صدا بزند تمام عناصر موجود در collection (مجموعه) C در آن درج می شود.	<code>public boolean addAll(Collection c)</code>	۲
برای حذف کردن یک عنصر از collection استفاده می شود.	<code>public boolean remove(Object element)</code>	۳
<p>برای حذف همه عناصری از یک collection که در collection (مجموعه) C نیز وجود دارد استفاده می شود. به عبارت دیگر اگر یک collection این متد را صدا زد، عنصرهای موجود در آن collection که با collection (مجموعه) C شباهت دارد حذف می شود. مثلا فرض کنید دو collection (مجموعه) به صورت زیر داریم:</p> <pre>a=["ali","sara"] b=["ali","maryam"]</pre> <p>حال دستور زیر را داریم:</p> <pre>b.removeAll(a)</pre> <p>دستور بالا چکار می‌کند؟ عناصری از مجموعه b که در مجموعه a موجود است حذف می‌کند. با دستور بالا عنصر "ali" از مجموعه b به دلیل این که در مجموعه a وجود دارد حذف می‌شود، و مقدار مجموعه b به صورت زیر خواهد بود:</p> <pre>b=["maryam"]</pre> <p>نکته: عناصر collection ای که این متد را صدا می‌زند حذف می‌شود.</p>	<code>public boolean removeAll(Collection c)</code>	۴

Core Java

آموزش ساده و آسان جاوا

<p>این متد کاربردش عکس متد removeAll است. این متد عناصری از یک collection که با collection (مجموعه) C شباهتی ندارد، حذف می کند. مثلا فرض کنید دو collection (مجموعه) به صورت زیر داریم:</p> <pre>a=["ali","sara"] b=["ali","maryam"]</pre> <p>حال دستور زیر را داریم:</p> <pre>b.retainAll(a);</pre> <p>دستور بالا چکار می‌کند؟ عناصری از مجموعه b که با مجموعه a شباهتی ندارد حذف می کند. با دستور بالا عنصر "maryam" از مجموعه b به دلیل این که در مجموعه a وجود ندارد حذف می شود، و مقدار مجموعه b به صورت زیر خواهد بود:</p> <pre>b=["ali"]</pre> <p>نکته: عناصر collection ای که این متد را صدا می زند حذف می شود.</p>	<pre>public boolean retainAll(Collection c)</pre>	۵
<p>این متد مجموع تعداد عناصر موجود در یک collection را برمی گرداند.</p>	<pre>public int size()</pre>	۶
<p>برای حذف کل عناصر موجود در یک collection استفاده می شود.</p>	<pre>public void clear()</pre>	۷
<p>برای جست و جوی یک عنصر در collection مورد استفاده قرار می گیرد. این متد چک می کند که آیا عنصر مورد نظر در collection وجود دارد یا خیر. اگر وجود داشت مقدار true و در غیر این صورت مقدار false برمی گرداند.</p>	<pre>public boolean contains(Object element)</pre>	۸

Core Java

آموزش ساده و آسان جاوا

<p>چک می کند که آیا همه عناصر موجود در collection (مجموعه) در C در collection که این متد را صدا زده است وجود دارد یا خیر. اگر وجود داشت مقدار true و در غیر این صورت مقدار false را برمی گرداند.</p> <p>مثلا فرض کنید دو collection (مجموعه) به صورت زیر داریم:</p> <pre>a=["ali","sara"] b=["ali","sara","maryam"]</pre> <p>حال دستور زیر را داریم:</p> <pre>b.containsAll(a);</pre> <p>دستور بالا چکار می‌کند؟ چک می‌کند که آیا همه عناصر موجود در مجموعه a در مجموعه b وجود دارد یا خیر؟ همان طور که مشاهده می کنید همه عناصر مجموعه a در مجموعه b نیز وجود دارد پس این متد مقدار true برمی گرداند.</p>	<pre>public boolean containsAll(Collection c)</pre>	۹
<p>یک iterator برمی گرداند. با استفاده از iterator برگردانده شده می توانیم یک collection را پیمایش کنیم.</p> <p>اصلا نگران درک کاربرد این متدها نداشته باشید همه رو با مثال کدنویسی توضیح خواهیم داد.</p>	<pre>public Iterator iterator()</pre>	۱۰
<p>برای تبدیل collection به آرایه استفاده می شود.</p>	<pre>public Object[] toArray()</pre>	۱۱
<p>خالی بودن یک collection را چک می کند. اگر collection خالی بود مقدار true و در غیر این صورت مقدار false را برمی گرداند.</p>	<pre>public boolean isEmpty()</pre>	۱۲
<p>برابری دو collection را بررسی می کند. اگر دو collection برابر بودند مقدار true و در غیر این صورت مقدار false را برمی گرداند.</p>	<pre>public boolean equals(Object element)</pre>	۱۳

Core Java

آموزش ساده و آسان جاوا

نکته مهم: شرط برابری دو collection ، علاوه بر شباهت همه عناصر باید ترتیب قرار گیری عناصر در هر دو collection نیز شبیه هم باشد.

مثلا فرض کنید دو collection (مجموعه) به صورت زیر داریم:

```
a=["ali","sara"]
```

```
b=["ali","sara"]
```

دو مجموعه a و b برابر هستند.

اما:

```
c=["ali","sara"]
```

```
d=["sara","ali"]
```

دو مجموعه c و d برابر نیستند، چرا؟ با وجود شباهت عناصر هر دو collection ، به دلیل این که ترتیب قرار گیری عناصر شبیه هم نیست، دو collection برابر نیستند.

hash code مجموعه ای (collection) که این متد را صدا زده است را برمی گرداند.

```
public int hashCode()
```

۱۴

این جلسه اول آموزش collection ها در جاوا می باشد. ما کاربرد متدهای موجود در اینترفیس Collection را بررسی کردیم. در جلسات بعدی به صورت مثال کدنویسی ، کاربرد هر کدام از متدهای موجود در جدول را بررسی می کنیم. پس اصلا نگران درک کاربرد این متدها نباشید.

Core Java

آموزش ساده و آسان جاوا

اینترفیس (Iterator interface)

اینترفیس Iterator (interface) امکانات **پیمایش** عناصر collection ها را فقط در جهت **رو به جلو** فراهم کرده است. همان طور که برای پیمایش عناصر موجود در یک آرایه از یک حلقه for استفاده می کردیم برای پیمایش عناصر collection ها از امکاناتی که اینترفیس Iterator برای ما فراهم کرده ، استفاده می کنیم. کاربرد این متد را در آموزش های بعدی خواهید دید.

یکی از راه های پیمایش collection ها ، استفاده از امکانات اینترفیس Iterator می باشد.

نکته

متدهای اینترفیس (Iterator interface)

تنها سه متد در اینترفیس Iterator وجود دارد که در جدول زیر جمع آوری کرده ایم:

شماره	متد	توضیح
۱	public boolean hasNext()	اگر بعد از عنصر فعلی ، عنصری وجود داشت مقدار true برمی گرداند ، در غیر این صورت مقدار false برمی گرداند.
۲	public Object next()	عنصر بعدی رو برمی گرداند. اگر عنصر بعدی وجود نداشت استثنای NoSuchElementException رخ می دهد.
۳	public void remove()	آخرین عنصر را حذف می کند. به ندرت از این متد استفاده می شود.

Core Java

آموزش ساده و آسان جاوا

هدف از این جلسه آموزشی آشنایی اولیه با collection ها در جاوا است، ما در این جلسه آموزشی از مثال کدنویسی استفاده نکردیم، خب ممکنه سوالات و ابهاماتی در مورد کاربرد مفاهیمی که توضیح دادیم وجود داشته باشد. اصلا نگران نباشید چرا که در جلسات آینده گام به گام اینترفیس ها و کلاس های موجود در فریم ورک collection جاوا را بررسی خواهیم کرد، و کاربرد همه این مفاهیم توضیح داده شده در قالب مثال مرور خواهد شد.

سعی ما تهیه بهترین و با کیفیت ترین جلسات آموزشی هست....

پیروز و موفق باشید

Core Java

آموزش ساده و آسان جاوا

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

www.JAVAPRO.ir

آموزش جاوا SE را با تجربه شفصی و به زبان خودمونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دفل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاواالایک به دور از افلاق حرفه ای ست و حرام می باشد.