Core Java

آموزش ساده و آسان جاوا

به نام خدا

تقدیم به هموطنان عزیزم

جاوا را با لذت یاد بگیرید!

Core Java

آموزش ساده و آسان جاوا

آموزش زبان برنامه نویسی جاوا

Collections در جاوا

موضوع: اینترفیس List

جلسه: پنجم

نویسنده : رحمان زارعی

جاوا را ساده،آسان و شیرین بنوشید!!!

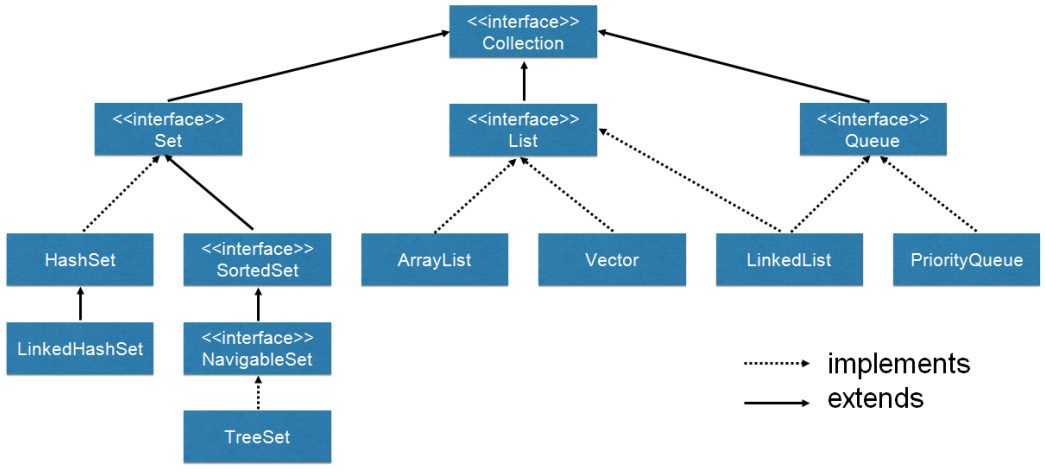
Core Java

آموزش ساده و آسان جاوا

سلام. در یک روز پاییزی در خدمت شما هستم.در این جلسه آموزشی جاوا قصد داریم یکی از اینترفیس های موجود در فریم ورک Collection را بررسی کنیم.

List Interface

طبق نمودار موجود در تصویر(۱) اینترفیس List فرزند اینترفیس Collection در جاوا می باشد.List دارای متدهایی است که از طریق آن ها می توان عناصر را بر اساس ایندکس هایشان (شماره خانه شان) درج یا حذف کرد.



تصویر(۱)

- چرا اینترفیس List فرزند اینترفیس Collection می باشد؟ زیرا اینترفیس List اینترفیس Collection را extends کرده است.
- اینترفیس List توسط کلاس های ArrayList ، LinkedList ، Vector وStack پیاده سازی یا implements شده است. به دلیل این که List اینترفیس می باشد ، هنگام شی سازی از آن می توانیم از سازنده های کلاس های ArrayList، LinkedList، Vector وStack کمک بگیریم.

Core Java

آموزش ساده و آسان جاوا

ایجاد اشیا از List :

همان طور که می دانید List یک اینترفیس است و ما نمی توانیم مستقیم از آن شی ایجاد کنیم. به روش های زیر می توانیم از یک List شی ایجاد کنیم:

```
List<String> a = new ArrayList<String>();
List<String> b = new LinkedList<String>();
List<String> c = new Vector<String>();
List<String> d = new Stack<String>();
```

- کافی است که نوع شی از نوع اینترفیس List و سازنده آن از یکی از کلاس های ArrayList، LinkedList ، Vector و Stack باشد.
- ما در جلسات آینده کلاس های Vector و Stack را بررسی خواهیم کرد. البته شما هم باید پی گیر و مطالبه گر باشید ☺ ☺
- تگ <> کنار نام List برای مشخص کردن نوع عناصری است که قراره در List ما ذخیره شوند. این نوع می تواند عدد، رشته، نام کلاسی خاص و.. باشد.
- ما به روش معمولی از List شی ایجاد می کنیم با این تفاوت که سازنده آن ، از یکی کلاس های ArrayList، LinkedList، Vector و Stack می باشد، مثلا:

```
List a = new ArrayList();
```

حالا نوع عناصر List را میان دو تگ <> کنار نام List و نام سازنده آن قرار می دهیم.

```
List<String> a = new ArrayList<String>();
```

برای استفاده از List در برنامه خود باید پکیج زیر را بالای سر کلاس خود import کنید:

```
import java.util.List;
```

همچنین در صورت استفاده از سازنده یکی از کلاس ArrayList، LinkedList ، Vector و Stack باید پکیج های مربوط آن ها را نیز در برنامه خود import کنید:

Core Java

آموزش ساده و آسان جاوا

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.Stack;
import java.util.Vector;
```

متدهای اینترفیس List :

شماره	متد	توضیح
۱	`void add(int index,Object element)`	یک عنصر را در خانه مشخص (index) از لیست درج می کند.
۲	`boolean addAll(int index,Collection c)`	همه عناصر موجود در Collection (مجموعه) c را از خانه index ام به بعد لیست اضاف می کند. کلاس های ArrayList و LinkedList و... یک Collection حساب می شوند.
۳	`object get(int index)`	عنصر ذخیره شده در خانه index ام لیست را برمی گرداند.
۴	`object set(int index,Object element)`	عنصر مورد نظر را به خانه index ام لیست اختصاص می دهد.
۵	`object remove(int index)`	عنصر مورد نظر در خانه index ام لیست را حذف می کند. مقدار عنصر را نیز بعد از حذف به ما پس می دهد.
۶	`ListIterator listIterator()`	برای پیمایش یک لیست از آن استفاده می کنیم
۷	`ListIterator listIterator(int index)`	برای پیمایش یک لیست از خانه index ام به بعد استفاده می شود. مثلا قصد دارید بجای خانه صفرم ، از خانه سوم به بعد شروع به پیمایش یک لیست کنید کافیست مقدار پارامتر این متد را ۳ قرار دهید.

Core Java

آموزش ساده و آسان جاوا

Example:

```
package list;

import java.util.*;

public class ListExample {
	public static void main(String args[]) {
		List<String> al = new ArrayList<String>();
		al.add("hasan");
		al.add("ali");
		al.add("sara");
		al.add(1, "maryam");
		System.out.println("Element at 2nd position: " + al.get(2));
		for (String s : al) {
			System.out.println(s);
		}
	}
}
```

خروجی (output):

```
Element at 2nd position: ali
hasan
maryam
ali
sara
```

توضیحات:

```
List<String> al = new ArrayList<String>();
```

- یک شی از List با استفاده از سازنده کلاس ArrayList ایجاد کرده ایم. عناصری که لیست ما می تواند ذخیره کند از نوع String می باشد.

```
al.add("hasan");
al.add("ali");
al.add("sara");
```

- متد add عنصر مورد نظر را به لیست اضاف می کند.به گونه ای که هر عنصر بعد از آخرین عنصر موجود در لیست قرار می گیرد. در اینجا عنصر "hasan" در خانه صفرم و عنصر "ali" در خانه یکم و عنصر "sara" در خانه بعد از عنصر "ali" یعنی خانه دوم قرار گرفته است.

Core Java

آموزش ساده و آسان جاوا

```
al.add(1, "maryam");
```

این دستور عنصر "maryam" را به خانه یکم اضاف می کند، همان طور که می دانید در خانه یکم عنصر "ali" قرار داشت، با این کار عنصر "ali" جابه جا شده و به خانه دوم منتقل میشه و عنصر "sara" نیز به خانه سوم منتقل میشه و "maryam" در خانه یکم قرار می گیرد.

```
System.out.println("Element at 2nd position: " + al.get(2));
```

- عنصر مورد نظر در خانه دوم لیست را برمی گرداند.

```
for (String s : al) {
                System.out.println(s);
            }
```

- برای پیمایش لیست از حلقه for-each استفاده کرده ایم. این حلقه رو در جلسات قبل توضیح داده ایم.

ListIterator Interface

اینترفیس ListIterator برای پیمایش رو به جلو و رو به عقب عناصر یک لیست استفاده می شود.

اینترفیس ListIterator ، اینترفیس Iterator را extends کرده است.

برای استفاده از اینترفیس ListIterator ، باید پکیج زیر را import کنید:

```
import java.util.ListIterator;
```

متدهای اینترفیس ListIterator :

Core Java

آموزش ساده و آسان جاوا

شماره	متد	توضیح
۱	boolean hasNext()	وقتی که در حال پیمایش رو به جلو لیست هستیم، این متد چک می کند آیا عنصری بعد از عنصر فعلی وجود دارد یا خیر. در صورت وجود مقدار true برمی گرداند. برای بررسی شرط وجود عنصر در لیست از این متد استفاده می شود.
۲	Object next()	عنصر بعد از عنصر فعلی را برمی گرداند.
۳	boolean hasPrevious()	وقتی که در حال پیمایش رو به عقب لیست هستیم، این متد چک می کند که ایا عنصری قبل از عنصر فعلی وجود دارد یا خیر. (در کل برای چک کردن وجود عنصر در لیست می باشد) برای پیمایش معکوس یک لیست از این متد استفاده می کنیم. منظور از پیمایش معکوس این است که از آخر به اول لیست را پیمایش می کنیم.
۴	Object previous()	عنصر قبل از عنصر فعلی را برمی گرداند.

- روش استفاده از امکانات اینترفیس ListIterator ، برای پیمایش یک لیست را در مثال زیر می توانید ببینید:

Example:

```
package list;

import java.util.*;

public class TestCollection8 {
      public static void main(String args[]) {
            List<String> al = new ArrayList<String>();
            al.add("hasan");
            al.add("ali");
            al.add("sara");
            al.add(1, "maryam");
            System.out.println("element at 2nd position: " + al.get(2));
            ListIterator<String> itr = al.listIterator();
            System.out.println("traversing elements in forward direction...");
            while (itr.hasNext()) {
                  System.out.println(itr.next());
            }
```

```
            System.out.println("traFversing elements in backward direction...");
            while (itr.hasPrevious()) {
                  System.out.println(itr.previous());
            }
      }
}
```

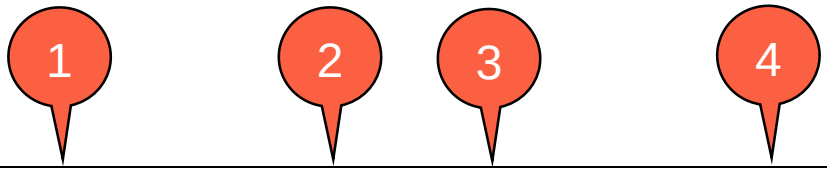
خروجی (output):

```
element at 2nd position: ali
traversing elements in forward direction...
hasan
maryam
ali
sara
traFversing elements in backward direction...
sara
ali
maryam
hasan
```

توضیحات:

```
ListIterator<String> itr = al.listIterator();
```

برای پیمایش لیست خود با استفاده از اینترفیس ListIterator ، به روش بالا از ListIterator شی ایجاد می کنیم.




```
ListIterator<String> itr = al.listIterator();
```

۱. نام اینترفیس ListIterator را تایپ می کنیم.

۲. نوع عناصر لیستی که قراره پیمایش شوند را بین دو تگ <> قرار می دهیم. نوع عناصر لیست ما از نوع String بود به همین خاطر <String> را در کنار نام اینترفیس ListIterator قرار داده ایم.

Core Java

آموزش ساده و آسان جاوا

۳. یک نام به دلخواه برای پیمایشگر خود انتخاب می کنیم.

۴. با استفاده از شی از نوع List متد listIterator() صدا می زنیم.

- قالب کلی ایجاد یک پیمایشگر ListIterator به صورت زیر است:

```
ListIterator< 1 > 2 = 3 .listIterator();
```

۱. تعیین نوع عناصری که قرار پیمایش شود.

۲. انتخاب یک نام دلخواه

۳. قرار دادن شی از نوع لیستی که قراره پیمایش شود.

حالا ایجاد یک حلقه برای پیمایش رو به جلو (شروع پیمایش از ابتدا تا انتهای لیست):

```
while (itr.hasNext()) {
            System.out.println(itr.next());
        }
```

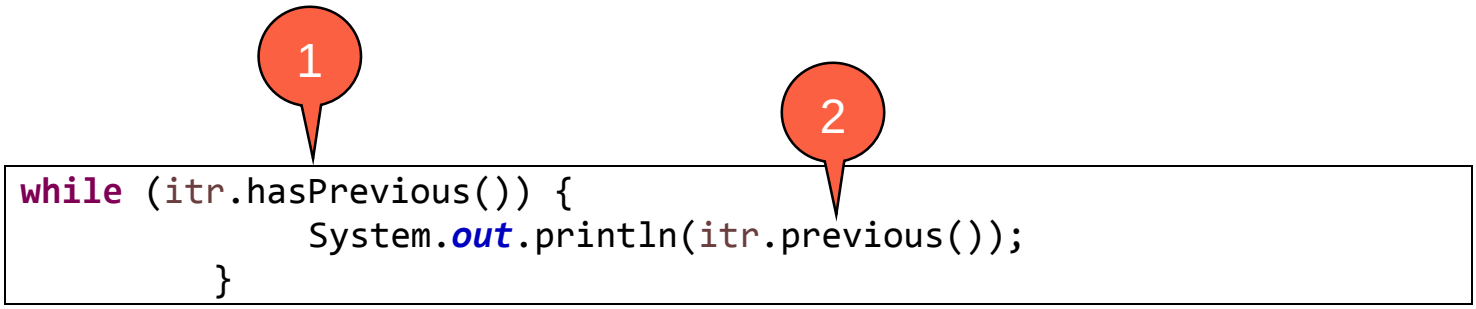
۱. شرط حلقه، چک می کند که آیا عنصری بعد از عنصر فعلی وجود دارد یا خیر. در کل برای بررسی وجود عنصر در لیست می باشد و این که به انتهای لیست رسیدیم یا خیر.

۲. با هر بار صدا زدن متد next عنصر بعد از عنصر فعلی برگردانده و سپس چاپ می شود.

Core Java

آموزش ساده و آسان جاوا

ایجاد یک حلقه برای پیمایش رو به عقب لیست (شروع پیمایش از انتها تا ابتدای لیست):




```
while (itr.hasPrevious()) {
                System.out.println(itr.previous());
          }
```

۱. بررسی می کند ، قبل از عنصر فعلی ، عنصری دیگر وجود دارد یا خیر. در کل چک می کند به ابتدای لیست رسیدیم یا خیر.

۲. این دستور عنصر قبل از عنصر فعلی را به ما می دهد و مقدارش را چاپ می کند.

پایان توضیحات مثال

متد List subList(int fromIndex,int toIndex) :

این متد بخشی از عناصر یک لیست را از خانه fromIndex تا toIndex برمی گرداند. عناصر برگردانده شده در قالب یک list می باشد.

توجه کنید که بازه ما به صورت [fromIndex,toIndex) می باشد ، یعنی هنگام برگرداندن بخشی از عناصر لیست، ایندکس fromIndex شامل می شود اما toIndex شامل نمی شود.

Example:

```
package list;

// Java program to demonstrate subList operation
// on List interface.
import java.util.*;

public class ListDemo {
      public static void main(String[] args) {
            // Type safe array list, stores only string
```

```
        List<String> l = new ArrayList<String>(5);

        l.add("javapro");
        l.add("javalike");
        l.add("java");
        l.add("like");
        l.add("tutorial");

        List<String> range = new ArrayList<String>();

        // return List between 2nd(including)
        // and 4th element(excluding)
        range = l.subList(2, 4);

        System.out.println(range); //[java, like]
    }
}
```

خروجی (output):

```
[java, like]
```

- در این مثال عناصر موجود در بازه [2,4) لیست را برگردانده ایم. بازه [2,4) شامل خانه های دوم و سوم لیست می شود و شامل خانه چهام نمی شود.

حتما برای تسلط بر این مفاهیم باید مثال های متعدد ببینید و حل کنید.

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 شخصی 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.