

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 Collections

موضوع: تفاوت بین ArrayList و LinkedList

جلسه: چهارم

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

سلام. امروز عصر در یک روز بارونی قصد دارم تفاوت بین کلاس ArrayList و کلاس LinkedList رو براتون بگم 😊

تفاوت بین ArrayList و LinkedList

ArrayList و LinkedList هر دو از کلاس های موجود در فریم ورک Collection در جاوا می باشند. هر دو، اینترفیس list را implements کرده اند.

کلاس ArrayList لیستی را ایجاد می کند که در داخل یک آرایه پویا (dynamic array) ذخیره می شود که اندازه آن با توجه به اضافه یا کم شدن عناصر، بزرگ تر یا کوچک تر می باشد.

LinkedList لیستی را ایجاد می کند که در داخل یک لیست پیوندی دو طرفه (doubly linked list) ذخیره می شود.

هر دو کلاس برای ذخیره عناصر در لیست استفاده می شوند، اما عمده تفاوت بین کلاس های ArrayList و LinkedList این است که ArrayList با توجه به این که یک ساختمان داده مبتنی بر index می باشد (یعنی می توانیم بر اساس شماره خانه هر عنصر ، به عنصر مورد نظر دسترسی پیدا کنیم)، اجازه دسترسی تصادفی (random) به عناصر موجود در لیست خود را می دهد.

از سوی دیگر LinkedList اجازه دسترسی تصادفی به عناصر خود را نمی دهد، چرا؟ به دلیل این که در LinkedList ایندکس (index) برای دسترسی مستقیم به عناصر وجود ندارد و تنها از طریق پیمایش LinkedList می توان به عناصر موجود در آن دسترسی پیدا کرد.

تفاوت های میان دو کلاس ArrayList و LinkedList را در جدول زیر جمع آوری کرده ایم:

شماره	LinkedList	ArrayList
۱	LinkedList اجازه دسترسی تصادفی به عناصر موجود در لیست را نمی دهد	ArrayList اجازه دسترسی تصادفی به عناصر موجود در لیست را می دهد.
۲	ساختمان داده داخلی که برای ذخیره عناصر استفاده می شود یک لیست پیوندی دو طرفه (doubly linked list) است.	ساختمان داده داخلی که برای ذخیره عناصر استفاده می شود یک آرایه پویا (dynamic array) است.

Core Java

آموزش ساده و آسان جاوا

ArrayList کلاس AbstractList را extends یا به ارث برده است.	LinkedList کلاس AbstractSequentialList را extends کرده است.	۳
کلاس AbstractList نیز اینترفیس List را implements کرده است. به دلیل این که کلاس ArrayList کلاس AbstractList به ارث برده و کلاس AbstractList نیز اینترفیس List را implements کرده است. ، پس می توان گفت که کلاس ArrayList اینترفیس List را implements کرده است	LinkedList علاوه بر اینترفیس list اینترفیس های Deque و Queue را implements کرده است. ما در جلسات آینده در مورد ها Deque و Queue بحث خواهیم کرد.	۴
دسترسی به عناصر موجود در لیست ArrayList سریع تر است.	دسترسی به عناصر موجود در لیست LinkedList کندتر است.	۵
دستکاری عناصر موجود در لیست ArrayList کندتر است.	دستکاری به عناصر موجود در لیست LinkedList سریع تر است.	۶
ArrayList فقط می تواند به عنوان یک list (لیست) عمل کند، زیرا تنها اینترفیس List را implements کرده است.	LinkedList هم می تواند به عنوان یک list (لیست) و هم یک Queue (صف) عمل کند. زیرا هر دو اینترفیس List و Queue را implements کرده است.	۷
ArrayList برای ذخیره و دسترسی به داده ها بهتر است.	LinkedList برای دستکاری داده ها بهتر است. منظور از دستکاری ، عملیات حذف یا اضافه کردن عناصر در لیست می باشد.	۸

نتیجه:

هنگام عملیات دستکاری، نظیر اضافه یا حذف کردن عناصر در لیست LinkedList مناسب تر است. اگر قصد جست و جوی مکرر در لیست را دارید، ArrayList بهترین انتخاب است، چرا که در دسترسی به عناصر لیست بهتر عمل می کند.

Core Java

آموزش ساده و آسان جاوا

یک مثال ساده از ArrayList و LinkedList را در زیر می توانید ببینید:

Example:

```
package linkedList;

import java.util.*;

class TestArrayLinked {
    public static void main(String args[]) {

        List<String> al = new ArrayList<String>();// creating arraylist
        al.add("Ravi");// adding object in arraylist
        al.add("Vijay");
        al.add("Ravi");
        al.add("Ajay");

        List<String> al2 = new LinkedList<String>();// creating linkedlist
        al2.add("James");// adding object in linkedlist
        al2.add("Serena");
        al2.add("Swati");
        al2.add("Junaid");

        System.out.println("arraylist: " + al);
        System.out.println("linkedlist: " + al2);
    }
}
```

خروجی (output):

```
arraylist: [Ravi, Vijay, Ravi, Ajay]
linkedlist: [James, Serena, Swati, Junaid]
```

- لازم به ذکر است که ما کلاس های ArrayList و LinkedList را در جلسات گذشته به صورت جداگانه بررسی کرده ایم.

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 شفقی 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.