

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: کلاس HashSet

جلسه: ششم

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 قصد داریم کلاس HashSet یکی از کلاس های موجود در فریم ورک Collection در جاوا رو بررسی کنیم.

کلاس HashSet کلاس AbstractSet را extends و اینترفیس Set را implements کرده است. HashSet برای ایجاد یک مجموعه ای (collection) است، که از یک جدول hash برای ذخیره سازی استفاده می کند.

- جدول hash یا hash table یک ساختمان داده برای جست و جو و ذخیره سازی اطلاعات است.

نکات مهم در مورد کلاس HashSet :

- عناصر موجود در HashSet منحصر به فرد هستند، به عبارت دیگر HashSet نمی تواند عناصر تکراری داشته باشد.
- HashSet برای ذخیره سازی عناصر از مکانیزمی به نام درهم سازی (hashing) استفاده می کند.
- اگر در یک HashSet بیش از یک مقدار null وارد کنیم، HashSet تنها یک مقدار null برای ما برمی گرداند.

تفاوت بین List و Set :

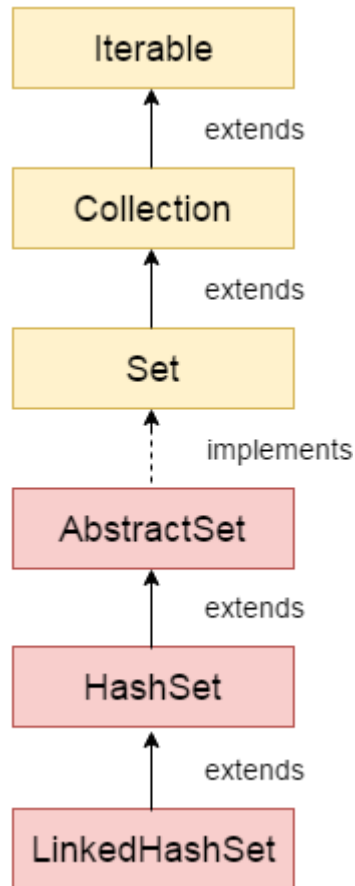
List می تواند عناصر تکراری داشته باشد اما Set عناصر یکتا و غیر تکراری می گیرد.

سلسله مراتب کلاس HashSet :

طبق تصویر (۱) ، کلاس HashSet کلاس AbstractSet که اینترفیس Set را implements کرده را extends یا به ارث برده است. اینترفیس Set اینترفیس Collection را به ارث برده و اینترفیس Collection اینترفیس Iterable را extends یا به ارث برده است به عبارت دیگر اینترفیس Set فرزند اینترفیس های Collection و Iterable می باشد.

Core Java

آموزش ساده و آسان جاوا



تصویر (۱)

Core Java

آموزش ساده و آسان جاوا

پکیج مورد نیاز برای تعریف کلاس HashSet :

برای استفاده از کلاس HashSet در برنامه خود باید پکیج زیر را بالا سر کلاس خود import کنید:

```
import java.util.HashSet;
```

سازنده های (Constructors) کلاس HashSet :

توضیح	سازنده	شماره
برای ایجاد یک HashSet پیشفرض استفاده می شود.	HashSet()	۱
برای ایجاد یک HashSet که توسط عناصر موجود در مجموعه C مقاردهی می شود. مجموعه C می تواند یکی از اینترفیس ها یا کلاس های موجود در فریم ورک Collection باشد.	HashSet(Collection c)	۲
برای ایجاد یک HashSet با ظرفیت مشخص استفاده می شود. ظرفیت یک HashSet با افزودن هر عنصر، به صورت اتوماتیک افزایش می یابد.	HashSet(int capacity)	۳

Core Java

آموزش ساده و آسان جاوا

متدهای (Methods) کلاس HashSet :

توضیح	متر	شماره
برای حذف کردن همه عناصر موجود در HashSet استفاده می شود.	void clear()	۱
اگر عنصر مورد نظر در HashSet وجود داشت ، مقدار true برمی گرداند.	boolean contains(Object o)	۲
برای افزودن عنصر مورد نظر به HashSet استفاده می شود. در صورت تکراری بودن عنصر یا به عبارت دیگر وجود داشتن عنصر در HashSet ، عنصر اضاف نمی شود.	boolean add(Object o)	۳
اگر هیچ عنصری در HashSet وجود نداشت، مقدار true برمی گرداند. به عبارت دیگر در صورت خالی بودن HashSet مقدار true برمی گرداند.	boolean isEmpty()	۴
اگر عنصر مورد نظر در HashSet وجود داشت ، آن را حذف می کند.	boolean remove(Object o)	۵
یک کپی از شی HashSet برمی گرداند. شی برگردانده شده از نوع کلاس Object هست که باید به HashSet آن را cast یا تبدیل کرد.	Object clone()	۶
برای پیمایش HashSet استفاده می شود.	Iterator iterator()	۷
تعداد عناصر موجود در HashSet را برمی گرداند.	int size()	۸

نکته: عناصر موجود در HashSet طبقه بندی نشده می باشد. به عبارت دیگر عناصر ذخیره شده در آن ترتیب خاصی ندارند و به صورت تصادفی در HashSet قرار می گیرند.

Core Java

آموزش ساده و آسان جاوا

Example:

```
package hashSet_javalike;

import java.util.HashSet;
import java.util.Iterator;

class TestCollection9 {
    public static void main(String args[]) {
        // Creating HashSet and adding elements
        HashSet<String> set = new HashSet<String>();
        set.add("mohammad");
        set.add("hasan");
        set.add("reza");
        set.add("mohammad");
        set.add("zahra");

        // Traversing elements
        Iterator<String> itr = set.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

خروجی (output): خروجی این برنامه امکان داره برای شما متفاوت باشد! چرا که عناصر در HashSet به صورت تصادفی و بدون ترتیب خاصی ذخیره می شوند.

```
hasan
zahra
mohammad
reza
```

توضیحات:

```
HashSet<String> set = new HashSet<String>();
```

یک HashSet که عناصری از نوع String می تواند ذخیره کند، ایجاد کرده ایم.

Core Java

آموزش ساده و آسان جاوا

```
1. set.add("mohammad");
2. set.add("hasan");
3. set.add("reza");
4. set.add("mohammad");
5. set.add("zahra");
```

- با استفاده از متد add عناصر مورد نظر را به HashSet خود اضافه کرده ایم. عنصر "mohammad" به دلیل تکراری بودن تنها یک بار اضافه می شود.

```
Iterator<String> itr = set.iterator();
while (itr.hasNext()) {
    System.out.println(itr.next());
}
```

- با استفاده از اینترفیس Iterator ، HashSet خود را پیمایش کرده ایم.

Example:

در مثال زیر ما یک کلاس Book داریم و از آن اشیا ساختیم، و سپس اشیا ایجاد شده از آن را در یک HashSet ذخیره کرده و در نهایت اطلاعات تمام اشیا Book ذخیره شده در HashSet را پیمایش و چاپ کرده ایم.

```
package hashSet_javalike;

import java.util.*;

class Book {
    int id;
    String name, author, publisher;
    int quantity;

    public Book(int id, String name, String author, String publisher,
                int quantity) {
        this.id = id;
        this.name = name;
        this.author = author;
        this.publisher = publisher;
        this.quantity = quantity;
    }
}

public class HashSetExample {
```


Core Java

آموزش ساده و آسان جاوا

```

public static void main(String[] args) {
    HashSet<Book> set = new HashSet<Book>();
    // Creating Books
    Book b1 = new Book(101, "Let us C", "Yashwant Kanetkar", "BPB", 8);
    Book b2 = new Book(102, "Data Communications & Networking",
"Forouzan",
        "Mc Graw Hill", 4);
    Book b3 = new Book(103, "Operating System", "Galvin", "Wiley", 6);
    // Adding Books to HashSet
    set.add(b1);
    set.add(b2);
    set.add(b3);
    // Traversing HashSet
    for (Book b : set) {
        System.out.println(b.id + " " + b.name + " " + b.author + " "
            + b.publisher + " " + b.quantity);
    }
}
}

```

خروجی (output):

```

101 Let us C Yashwant Kanetkar BPB 8
102 Data Communications & Networking Forouzan Mc Graw Hill 4
103 Operating System Galvin Wiley 6

```

Example:

```

package hashSet_javalike;

import java.util.HashSet;

public class HashSetExample {
    public static void main(String args[]) {
        // HashSet declaration
        HashSet<String> hset = new HashSet<String>();

        // Adding elements to the HashSet

```

Core Java

آموزش ساده و آسان جاوا

```

    hset.add("Apple");
    hset.add("Mango");
    hset.add("Grapes");
    hset.add("Orange");
    hset.add("Fig");
    // Addition of duplicate elements
    hset.add("Apple");
    hset.add("Mango");
    // Addition of null values
    hset.add(null);
    hset.add(null);

    // Displaying HashSet elements
    System.out.println(hset);
}
}

```

خروجی (output):

```
[null, Apple, Grapes, Fig, Mango, Orange]
```

- همان طور که می دانید اگر بیش از یک مقدار null به HashSet اضافه کنیم، تنها یکی از آنها در HashSet ذخیره می شود.

مثال از متد `clear()`:

Example:

```

package hashSet_avalike;

import java.util.*;

public class HashSetDemo1 {

    public static void main(String args[]) {

        HashSet<String> hashSetObject = new HashSet<String>();

        hashSetObject.add("I");
    }
}

```

Core Java

آموزش ساده و آسان جاوا

```
hashSetObject.add("Love");  
hashSetObject.add("Java");  
System.out.println("values in HashSet object " + hashSetObject);  
// Calling clear() mehthod  
hashSetObject.clear();  
System.out.println("values in HashSet object After using Clear method"  
+ hashSetObject);  
}  
}
```

خروجی (output):

```
values in HashSet object [Java, Love, I]  
values in HashSet object After using Clear method[]
```

مثال از متد Object clone() :

Example:

```
package hashSet_javalike;  
  
import java.util.*;  
  
public class HashSetDemo2 {  
  
    public static void main(String args[]) {  
  
        HashSet<String> hashSetObject1 = new HashSet<String>();  
  
        HashSet<String> hashSetObject2 = new HashSet<String>();  
  
        hashSetObject1.add("I");  
  
    }  
  
}
```

Core Java

آموزش ساده و آسان جاوا

```

hashSetObject1.add("Love");

hashSetObject1.add("Java");

hashSetObject2 = (HashSet) hashSetObject1.clone();

System.out.println("values in HashSet clone object " +
hashSetObject2);
    }
}

```

خروجی (output):

```
values in HashSet clone object [Java, I, Love]
```

توضیحات:

```
hashSetObject2 = (HashSet) hashSetObject1.clone();
```

- هنگام استفاده از متد clone برای کپی کردن شی از HashSet باید عمل casting در جاوا (تبدیل نوع) رو انجام دهید.

مثال از متد `boolean contains(Object o)` :

```

package hashSet_avalike;

import java.util.*;

public class HashSetDemo {

    public static void main(String args[]) {

        HashSet<String> hashSetObject = new HashSet<String>();

        hashSetObject.add("I");

        hashSetObject.add("Love");

        hashSetObject.add("Java");
    }
}

```

Core Java

آموزش ساده و آسان جاوا

```
if (hashSetObject.contains("I")) {  
    System.out.println("Hash Set Contains 'I'");  
}  
else {  
    System.out.println("Hash Set Do not Contains 'I'");  
}  
if (hashSetObject.contains("Iran")) {  
    System.out.println("Hash Set Contains 'Iran'");  
}  
else {  
    System.out.println("Hash Set Do not Contains 'Iran'");  
}  
}
```

خروجی (output):

```
Hash Set Contains 'I'  
Hash Set Do not Contains 'Iran'
```

مثال از متد `boolean isEmpty()`:

```
package hashSet_javalike;  
import java.util.*;  
public class HashSetDemo4 {  
    public static void main(String args[]) {
```

Core Java

آموزش ساده و آسان جاوا

```
HashSet<String> hashSetObject = new HashSet<String>();

hashSetObject.add("I");
hashSetObject.add("Love");
hashSetObject.add("Java");

// Checking whether hashset is empty
boolean flag = hashSetObject.isEmpty();

System.out.println("Is hash set empty ? " + flag);

// clearing hashset
hashSetObject.clear();

flag = hashSetObject.isEmpty();

System.out.println("Is hash set empty ? " + flag);
}
```

خروجی (output):

```
Is hash set empty ? false
Is hash set empty ? true
```

مثال از متد `boolean remove(Object o)`:

```
package hashSet_javalike;

import java.util.*;

public class HashSetDemo5 {

    public static void main(String args[]) {
```

Core Java

آموزش ساده و آسان جاوا

```
HashSet<String> hashSetObject = new HashSet<String>();  
hashSetObject.add("I");  
hashSetObject.add("Love");  
hashSetObject.add("Java");  
System.out.println("hash set values :" + hashSetObject);  
// calling remove method to remove "java"  
hashSetObject.remove("Java");  
System.out.println("hash set values after using remove method:"  
+ hashSetObject);  
}  
}
```

خروجی (output):

```
hash set values :[Java, Love, I]  
hash set values after using remove method:[Love, I]
```

مثال از متد `size()` :int

```
package hashSet_avalike;  
import java.util.*;  
public class HashSetDemo {  
    public static void main(String args[]) {  
        int size;  
        HashSet<String> hashSetObject = new HashSet<String>();
```

Core Java

آموزش ساده و آسان جاوا

```
hashSetObject.add("I");  
hashSetObject.add("Love");  
hashSetObject.add("Java");  
size = hashSetObject.size();  
System.out.println("Size of hash set is " + size);  
}  
}
```

خروجی (output):

```
Size of hash set is 3
```

- از مثال های متعدد، در زمینه کاربرد متدهای کلاس HashSet استفاده کردیم، گاهی برای یادگیری یک مفهوم برنامه نویسی نیاز به توضیح نیست و دیدن مثال کفایت می کند.

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.