ArrayList class Constructors

In previous post we saw the [introduction](http://data-structure-learning.blogspot.com/2015/08/arraylist-class-introduction-and-how-it.html) on ArrayList class and how it is [different](http://data-structure-learning.blogspot.com/2015/05/difference-between-arraylist-and-arrays.html) from arrays. We also saw [List](http://data-structure-learning.blogspot.com/2015/05/java-collections-part-5list-interface.html) interface and some details of [ArrayList](http://data-structure-learning.blogspot.com/2015/05/explaination-arraylist-class.html) class.

In this post we will learn about 3 different types of constructors provided in ArrayList class.

1. No-args constructor – This constructor is used to create a default list of size 10.
2. Initial Capacity as argument – This constructor takes *int* as argument which specifies initial capacity of list.
3. Collection as argument – This constructor constructs an ArrayList from the specified Collection provided as argument.

Above is the gist now let us see how to write these constructors, add some elements into it and print it out.

1. No-args constructor – As we just saw this constructor creates a default list with size 10. Below is the code uses default constructor and added few elements of type String into it.

/\*\*

\* ArrayList<> class has a no-args constructor.

\* This constructor creates an empty list with

\* initial capacity as 10.

\* \*/

**public** **static** **void** arraylistContructorNoArgs() {

List<String> names = **new** ArrayList<String>();

names.add("Ned");

names.add("Catelyn");

names.add("Rob");

names.add("Arya");

names.add("Sansa");

System.***out***.println(names);

}

/\*\*

\* Output

\* [Ned, Catelyn, Rob, Arya, Sansa]

\* \*/

1. Initial Capacity as Argument – In this type of constructor we give an int value to constructor and it creates the list of that size.

/\*\*

\* ArrayList<> class has a constructor that takes int as argument.

\* argument acts as initial capacity of ArrayList.

\* This constructor creates an empty list with

\* initial capacity as 10.

\* \*/

**public** **static** **void** arraylistConstructorInitialCapacity() {

List<String> names = **new** ArrayList<String>(20);

names.add("Ned");

names.add("Catelyn");

names.add("Rob");

System.***out***.println(names);

}

/\*\*

\* Output

\* [Ned, Catelyn, Rob]

\* \*/

1. Collection as argument – This constructor creates a list of Collection, in an order elements are returned by the Collection.

/\*\*

\* ArrayList<> class has a constructor that accepts

\* Collection as argument.

\* \*/

**public** **static** **void** arraylistConstructorCollectionArgs() {

List<String> days = **new** ArrayList<String>();

days.add("Monday");

days.add("Tuesday");

days.add("Wednesday");

List<String> names = **new** ArrayList<String>(days);

names.add("Ned");

names.add("Sansa");

System.***out***.println(names);

}

So that’s all on 3 constructors of ArrayList. In next post we will see 2 add() methods of ArrayList class.