Init arraylist with values

You can provide set.of or list.of Factory Method, since Java © 9 or Arrays.asList Factory Method to the ArrayList (Collection) Constructor to create and init At arrayList leapart in a line, you can use additional methods and additional After the creation time to initialize the walk of a arraylistlet through this tutorial to explore them in more detail initialized in a row with Java 9+ list.Of and Set.of provides set.of or list.of factory Method, from because Java 9+, the constructor ArrayList (Collection) to create and start a list of arrays in a row when creating @Test Public Void InitWithListoFandSeof () {list = ArrayList1 New arrayList (List.of (3, 1, 2)); assertthat (arraylist1) .Contains (3, 1, 2); List = ArrayList2 New arrayList (set.of (5, 4, 6)); assertthat (arraylist2) .Contains (5, 4, 6); } List.of allows duplicate items while Set.of not, throws illegally IllentaLaRgumentException Both list.of and Set.f not allow a null element, throw NullPointerException Initialize in a row with Arrays.asList provides Arrays.asList the Manufacturer ArrayList (Collection) To create and init a arrayList Arraylist in a row array.aslist accepts null and duplicate items @Test Public Void InitWitHirRaSaSList () {List Set = New ArrayList (Arrays.asList (3, 1, 2, null)); asserttthat (set) .Contains (3, 1, 2, null); } Initialization in a row from an existing collection Provide an existing collection to the constructor dell'arraylist (collection) to create and init in arrayList in a row @Test Public Void InitFrmansExistCollection () {List = ArrayList1 New arrayList (array . Aslist (3, 1, 2, NULL)); List = ArrayList2 New arrayList (arrayList1); assertthat (arraylist2) .Contains (3, 1, 2, null); } Initialize using additional methods and addonl You can use the Add method or addall to initialize a arrayList after the creation time @Test Public Void INITWITHADDANDADDALL () {List = arrayList1 New arrayList (set.of (1 , 2, 3)); List = ArrayList2 New arrayList (); arraylist1.add (1); arraylist1.add (1); // duplicate item is ignored arraylist1.add (null); // NULL element allows arrayList2.Addall (ArrayList1); asserttthat (arraylist2) .Contains (1, 2, 3, null); } The initialization of the double brace should not be used in practice, the initialization of the double brace can cause a memory leak because it creates anonymous classes with a hidden reference to the object instance of the owner @Test Public Void InitWithDoubleBrace () { List ArrayList = New ArrayList () {{add (3); Add (1); add (2); }} ;;; assertthat (arraylist) .Contains (1, 2, 3); } Initializes with Use this ability to improve performance When it is necessary to add a large number of elements with respect to the default capacity dell'arraylist (10) internally, a arrayList is supported by an array of fixed size. When its capacity is full, Java needs to create a new array with wider capabilities and copying old items that beginning Void Public ItWithCapacity () {List ArrayList = New ArrayList (1000); for (int i = 0; i