Cheatography

Java Array and ArrayList (OCA) Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19274/

Array Initialization

int [] a1; int a2[];	declarations
int [] a = new int[3];	a={0,0,0}
int [] a = {1,2,3};	initializer list
int [] a = new int[] {1,2,3};	initializer list
int a[] = new int[];	compile error
int [] a = new int[3] {1,2,3};	compile error

Array length

```
System.out.print(a.length); //3
```

Compare:length for Array, length() for String and size() for ArrayList

Loop through Array

```
for (int i=0; i<arr.length; i++)
{} //modify
for ( int a : arr) {...}
//access only
```

Array as method parameter

passing an array is passing the memory address, the change on array content will be persistent through caller and callee.

Array contents

```
int [] intArr={1,2,3}
int [] doubleArr=new double[3]
String [] objArr = new
String[3]; //object type
```

Array Modification

Array is fixed in length once initialized, remove an element will resize the array and move all the element on its right. This could be computationally expensive.

Arrays utilities

```
import java.util.Arrays;
Arrays.sort(arr);
Arrays.binarySearch(arr,item);
//if found, return index
//if not return -(potential
index -1)
//if array not sorted,result is
undefined
List list=Arrays.asList(arr);
Arrays.copyOfRange(Object[]
src,from,to)
```

toString() method

not overloaded, inherited from Object class directly

Variable arguments varargs

```
public static void main (String
[] args){}
public static void main
(String... args){}
//0 or more arguments
//similar to method overloading,
or arrays
//args are treated as args[]
```

Multiple dimension array

```
//initialization, first index
required
int [][] arr = new int[4][];
int [] arr[]= new int[5][];
int arr[][]= new int[6][]
int [] arr[][]= new int[5][][];
//3D
arr={{1,2,3},[4,5,6}};
//loop through
for (int i=0;i<arr.length;i++){
   for (int j=0;j<arr[i];j++)
{... }
}
for (int innerArr:arr){
  for (int num:innerArr){...}
}
```

ArrayList

import java.util.ArrayList;

Size dynamically changed at runtime with autoshift

last element indexed by size()-1

only contains reference type

Array vs ArrayList

Array	ArrayList
length predefined	size() variable
manual shift	Auto shift
faster	slower
primitive+Ref type	ref type only
ArrayIndexOutOfBoundaryException	IndexOutOfBoundaryException
equals() not overriden	overriden equals()
toString() not overriden	overriden toString()
Arrays.sort()	Collections.sort()
Arrays.binarySearch()	Collections.binarySearch()

Create ArrayList

```
ArrayList aL1 = new ArrayList();
ArrayList aL2 = new ArrayL-
ist(5);
ArrayList aL3 = new ArrayList-
(aL1);
ArrayList<String> aL4,aL5;
aL4=new ArrayList<String>();
//Java5
aL5 = new ArrayList<>(); //Java
7
ArrayList aL6 = Arrays.asLis-
t(arr);
```

IllegArgumentException // when primary type value assigned

Cheatography

Java Array and ArrayList (OCA) Cheat Sheet

by Jianmin Feng (taotao) via cheatography.com/79308/cs/19274/

Using ArrayList

add	Boolean add(E element)
	void add (int index, E element)
remove	Boolean remove(Object obj)
	E remove(int index)
set	E set(int index, E element)
size	int size()
	boolean isEmpty()
clear	void clear()
contains	boolean contains(Object obj)
equals	boolean equals(Object obj)

Wrraper class

To put primitive type to ArrayList, using wrapper class

Boolean,Byte,Short,Integer,Long,Float,Double,Character

Xxx.parseXxx(String) //convnert string to a primitive

Xxx.valueOf(String) //convert string to a wrapper class(Xxx)

NumberFormatException

Autoboxing

Auto primitive value > wrapper class obj

null allowed boxing into list, print ok,but unboxing is not allowed --> NullPointexception

Integer List is interest: int as index is precedent autoboxing. remove(1) will remove second element, not Integer(1)

Convert Between Array and List

```
1. toArray()
listObj.toArray(); //convert to obj arr
listObj.toArray(new String[0]);
//index= 0 will return a new string array,
//index >0 will reuse if fit, else return new one
2. AsList()
//get a list with fixed size
list =Arrays.asList(arrayObj)
list.remove(1): //UnsupportedOperation
List<String> args=Arrays.asList("one","two");//varargs
// get a new arrayList obj,
ArrayList<String> aL=new ArrayList(Arrays.asList("one","two"));
```

toArray() is list object method, return new array object

asList() is Arrays class static method, return the memory location as a list (same structure shared by array and list, content can be changed by both array and list,size still fixed)

By **Jianmin Feng** (taotao)
cheatography.com/taotao/

Not published yet.
Last updated 10th April, 2019.