Topics

- Arrays class in Java

Arrays class in Java

- Arrays class in <<java.util>> package is defined as follows
- This class contains the required methods for manipulating arrays such as sorting and searching
- All methods are <<static>> and hence can be called via class name.
- Arrays class is part of Java's Collection Framework

Arrays class: Important Methods

binarySearch()

static int	**binarySearch**(byte[] a, byte key)
static int	**binarySearch**(char[] a, char key)
static int	**binarySearch**(double[] a, double key)
static int	**binarySearch**(float[] a, float key)
static int	**binarySearch**(int[] a, int key)
static int	**binarySearch**(long[] a, long key)
static int	**binarySearch**(Object[] a, Object key)
static int	**binarySearch**(Object[] a, Object key, Comparator c)
static int	**binarySearch**(short[] a, short key)

Arrays class: Important Methods ….

equals()

static boolean	equals(boolean[] a, boolean[] a2)
static boolean	equals(byte[] a, byte[] a2)
static boolean	equals(char[] a, char[] a2)
static boolean	equals(double[] a, double[] a2)
static boolean	equals(float[] a, float[] a2)
static boolean	equals(int[] a, int[] a2)
static boolean	equals(long[] a, long[] a2)
static boolean	equals(Object[] a, Object[] a2)
static boolean	equals(short[] a, short[] a2)

Arrays class: Important Methods ….

fill()

static void	fill(boolean[] a, boolean val)
static void	fill(boolean[] a, int fromIndex, int toIndex, boolean val)
static void	fill(byte[] a, byte val)
static void	fill(byte[] a, int fromIndex, int toIndex, byte val)
static void	fill(char[] a, char val)
static void	fill(char[] a, int fromIndex, int toIndex, char val)
static void	fill(double[] a, double val)
static void	fill(double[] a, int fromIndex, int toIndex, double val)
static void	fill(float[] a, float val)

Arrays class: Important Methods ….

fill() ….

static void	fill(float[] a, int fromIndex, int toIndex, float val)
static void	fill(int[] a, int val)
static void	fill(int[] a, int fromIndex, int toIndex, int val)
static void	fill(long[] a, int fromIndex, int toIndex, long val)
static void	fill(long[] a, long val)
static void	fill(Object[] a, int fromIndex, int toIndex, Object val)
static void	fill(Object[] a, Object val)
static void	fill(short[] a, int fromIndex, int toIndex, short val)
static void	fill(short[] a, short val)

Arrays class: Important Methods ….

sort()

static void	sort(byte[] a)
static void	sort(byte[] a, int fromIndex, int toIndex)
static void	sort(char[] a)
static void	sort(char[] a, int fromIndex, int toIndex)
static void	sort(double[] a)
static void	sort(double[] a, int fromIndex, int toIndex)
static void	sort(float[] a)
static void	sort(float[] a, int fromIndex, int toIndex)
static void	sort(int[] a)
static void	sort(int[] a, int fromIndex, int toIndex)

Arrays class: Important Methods ….

sort()….

static void	sort(long[] a)
static void	sort(long[] a, int fromIndex, int toIndex)
static void	sort(Object[] a)
static void	sort(Object[] a, Comparator c)
static void	sort(Object[] a, int fromIndex, int toIndex)
static void	sort(Object[] a, int fromIndex, int toIndex, Comparator c)
static void	sort(short[] a)
static void	sort(short[] a, int fromIndex, int toIndex)

Arrays class: Example

```
import java.util.*;
class ArrayExample
{
public static void main(String args[])
{
int x[] = {10,6,8,20};

double data[]   = { 12.5,34.6,90.56,12.34,12.56};

float values[]  = { 10.45f,23.56f,12.67f};

double data1[]  = new double[10];

boolean flags[] = new boolean[5];

int x1[] = {10,6,8,20};
```

Import java.util package to use Arrays class

int array Size = 4 LB=0 UB =3

double array Size = 5 LB=0 UB =4

float arrays Size = 3 LB=0 UB =2

double array Size = 10 LB=0 UB =9

boolean array Size = 5 LB=0 UB =4

int array Size = 4 LB=0 UB =3

Arrays class: Example ….

```
System.out.println(Arrays.binarySearch(x,20));

Arrays.sort(x);

for(int i=0;i<x.length;i++)
System.out.print(x[i]+" ");
System.out.println();

Arrays.fill(data1,12.56);

for(int i=0;i<data1.length;i++)
System.out.print(data1[i]+" ");
System.out.println();
```

Prints index of 20 in x

3

Sorts elements of x

Prints Elements of x

6 8 10 20

Fills a single value 12.56 in all indexes of data1

Prints Elements of data1

12.56 12.56 12.56 12.56 12.56 12.56 12.56 12.56 12.56 12.56

Arrays class: Example ….

```
Arrays.fill(flags,2,5,true);
//Arrays.fill(flags,2,6,true);
for(int i=0;i<flags.length;i++)
System.out.print(flags[i]+" ");
System.out.println();

System.out.println(Arrays.equals(x,x1));

Arrays.sort(values);

for(int i=0;i<values.length;i++)
System.out.print(values[i]+" ");
System.out.println();

 }
 } // End of class
```

Fills true value in boolean array flags from index 2 to 4

ArrayIndexOutofBoundsException

false false true true true

Prints true or false whether x and x1 equals

Sorts the elements of arrays

10.45 12.67 23.56

Array : Examples

```
int a[] = { 10,8,6};
for(int i=0;i<a.length;i++)
System.out.println(a[i]);
```

```
10
8
6
```

```
int a[3] = { 10,8,6};
for(int i=0;i<a.length;i++)
System.out.println(a[i]);
```

```
']' expected
int a[3] = { 10,8,6};
      ^
1 error
```

```
//int table[][]={0,0,0,1,1,1};
//int table[2][3]={0,0,0,1,1,1};
```

WRONG

```
int table[ ][ ]={{0,0,0},{1,1,1}};
for(int i=0; i<table.length;i++)
{
for(int j=0;j<table[i].length;j++)
{
System.out.print(table[i][j]);}
System.out.println();
}
```

```
000
111
```

Thank You