**String class**

**=============================================================**

**Note:- String is a class which we can use to hold sequence of charecters.**

**Syntax:- String var=string value;**

**EX:- String str="Lava Kumar";**

**System.out.println("my string value is :"+str);**

**Note:- string will occupy number charecters in a string with multiplication of 16 bits.**

**EX:- String str="Lava Kumar";-->will occupy 9\*16 bits of memory.**

**===============================================================**

**Some of the important String methods:-**

**(1) length():- To find number of charecters in a given String.**

**EX:- String str ="LiveTech";**

**System.out.println("length of given string is :"+str.length()); //8**

**(2) charAt():- To read charecters from main String based on given Index.**

**NOTE:- Index starts with zero.**

**EX:- String str ="LiveTech"; // (01234567)**

**//to read "v"**

**System.out.println(str.charAt(2)); // v**

**(3) indexOf():- To find index of given charecters in main String.**

**EX:- String str ="LiveTech";**

**//to find index of "T"**

**System.out.println(str.indexOf("T")); // 4**

**(4) toUpperCase():- it converts given String into uppercase.**

**EX:- String str ="LiveTech";**

**//to convert into uppercase.**

**System.out.println(str.toUpperCase()); //LIVETECH**

**(5) toLowerCase():-it will converts given String into lowercase.**

**EX:- String str ="LiveTech";**

**System.out.println(str.toLowerCase()); // livetech**

**(6) equals():- To compare the Strings.**

**EX:- String str ="LiveTech";**

**System.out.println(str.equals("LiveTech")); //True**

**System.out.println(str.equals("Livetech")); // false**

**(7) equalsIgnoreCase():- To compare the Strings.**

**>>> It is case in sensitive**

**EX:- String str ="LiveTech";**

**System.out.println(str.equals("livetech")); // false**

**System.out.println(str.equalsIgnoreCase("livetech")); // true**

**(8) startsWith():- to compare prefix value of a string (i.e.,starting charecters).**

**EX:- String str ="LiveTech";**

**System.out.println(str.startsWith("live")); // false**

**System.out.println(str.startsWith("Live")); //true**

**(9) endsWith():- to check suffix value of a string (i.e.,end charecters).**

**EX:- String str ="LiveTech";**

**System.out.println(str.endsWith("Tech"));// true**

**System.out.println(str.endsWith("ch1")); // false**

**(10) contains():- to check subString availability in a main String.**

**EX:- String str ="LiveTech";**

**System.out.println(str.contains("Tech")); //true**

**System.out.println(str.contains("ve")); //true**

**System.out.println(str.contains("ve5")); //false**

**(11) concat():- to concat the Strings.**

**EX:- String str1 ="Live";**

**String str2 ="Tech";**

**System.out.println(str1.concat(str2)); // Livetech**

**System.out.println(str1+str2); // Livetech**

**(12) trim():- it will remove leading & Trailing spaces for a given String.**

**EX:- String str ="\*Live\*"; (\*)==> indicates SPACE**

**System.out.println(str.length()); // 1+4+1=>6**

**System.out.println(str.trim().length()); //4**

**(13) replace():- to replace substring in a main string.**

**EX:- String str ="LiveTech";**

**System.out.println(str.replace("e","@")); //Liv@T@ch**

**EX:- String str ="Sr Nagar";**

**System.out.println(str.length()); //8**

**System.out.println(str.trim().length()); //8**

**System.out.println(str.trim()); //Sr Nagar**

**System.out.println(str.replace("\*","")); //SrNagar**

**(14) isEmpty():- to check value in a variable is empty or not.**

**EX:- String str1="LiveTech";**

**String str2 ="\*";**

**System.out.println(str1.isEmpty()); //false**

**System.out.println(str2.isEmpty()); //true**

**(15) subString():- to read sub String from main String.**

**EX:- String str="LiveTech";**

**System.out.println(str.subString(4)); //Tech**

**System.out.println(str.subString(4,5)); //T**

**System.out.println(str.subString(4,6)); //Te**

**========================================================================**

**Arrays**

**Java array is an object which contains elements of a similar data type.We can store only a fixed set of elements in a Java array. Array in java is index-based, the first element of the array is stored at the 0 index.**

**public class fdfdfd {**

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

**int a[]={33,3,4,5};**//declaration, instantiation and initialization at the same time

//reading the data from the array and printing the elements by for loop

**for(int i=0;i<a.length;i++) {//length is the property of array**

**System.out.println(a[i]);**

**}**

//length is the predefined method to find the size of the array

//reading the data from the array and printing the elements by for each loop

**for(int element:a) {**

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

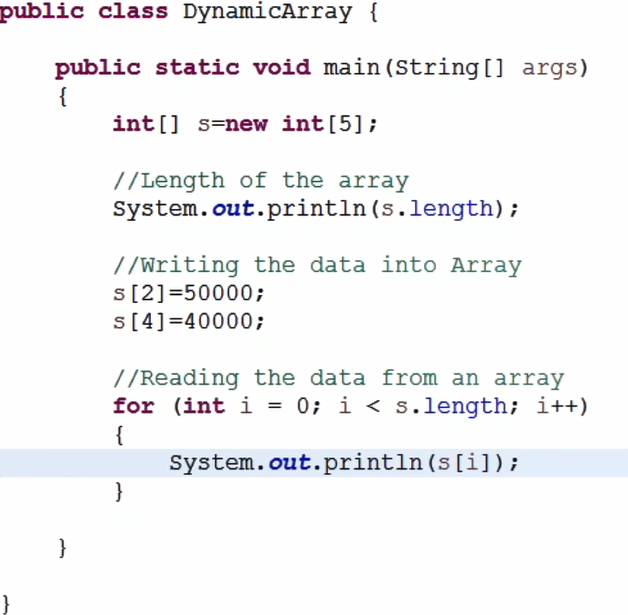
**}**

**}}**

Note: Java arrays are fixed in size ,so we cannot increase the size during runtime when once it is declared

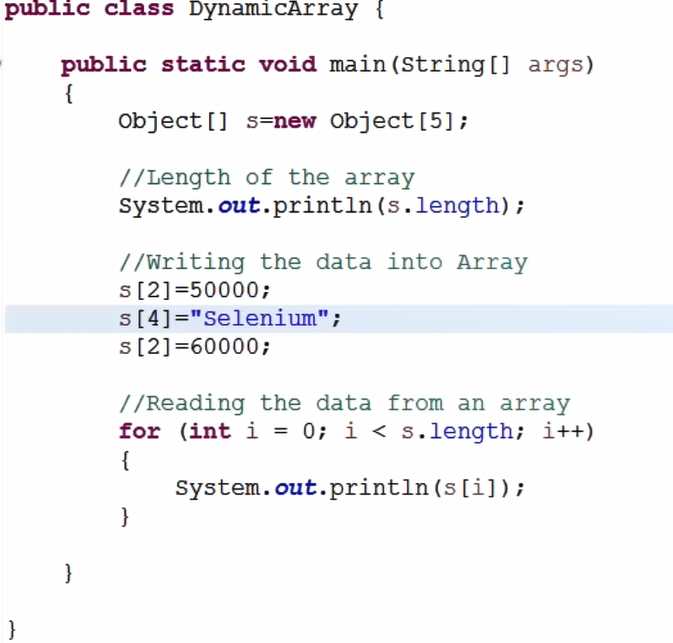
Note: The Java Virtual Machine (JVM) throws an ArrayIndexOutOfBoundsException if length of the array in negative, equal to the array size or greater than the array size while traversing the array.

//declaration/instantiation and initialization at different time

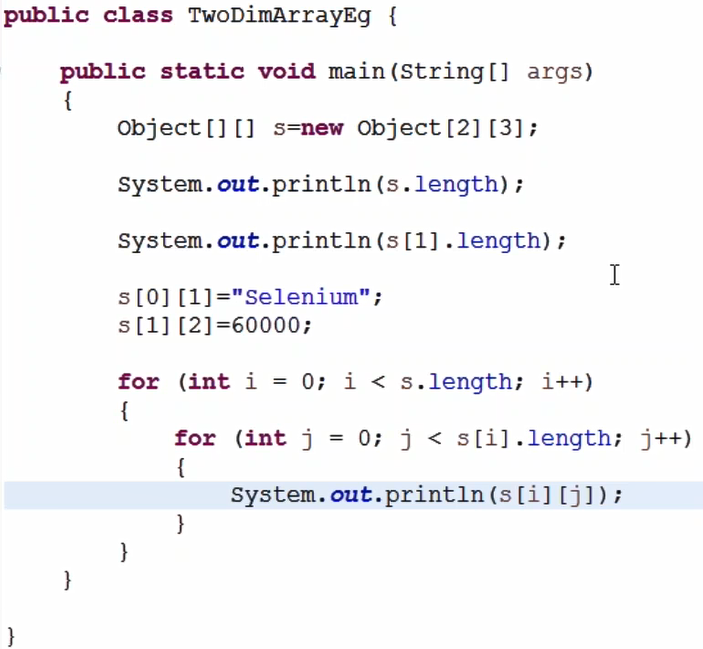


**Object Array**

Note: Object array is used to store multiple elements with different data types

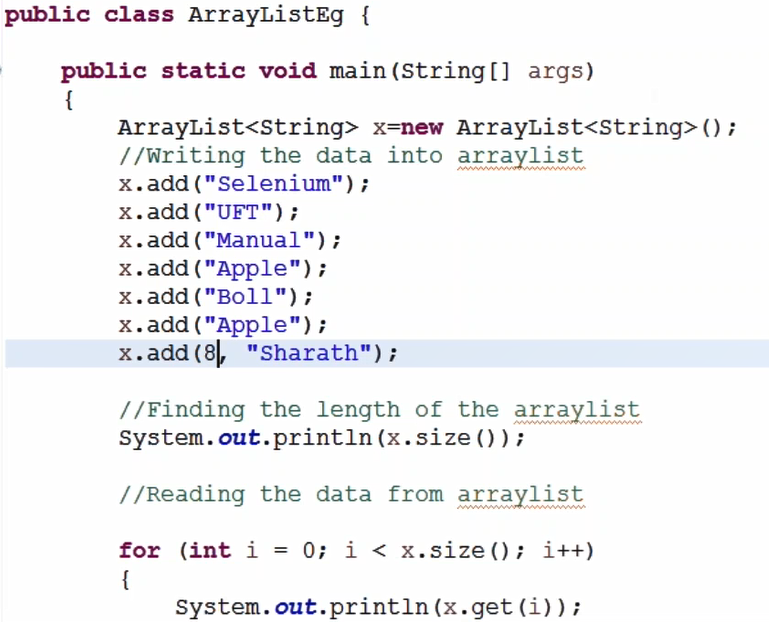


**Two Dimensional Arrays**



Note: with arrays we cannot increase the size and also there may be memory wastage because we cannot expect how many elements we store in the array before starting of the program to declare the length of the array

ArrayList will help us in this situation, in arraylist we can add any numbers of elements during runtime



**}}}**

Note:Java ArrayList class we need to add elements sequentially,it can contain duplicate elements,Java ArrayList class maintains insertion order,Java ArrayList allows random access because array works at the index basis,

//Traversing list through Iterator

Iterator itr=x.iterator();

**while**(itr.hasNext()){

System.***out***.println(itr.next());

}

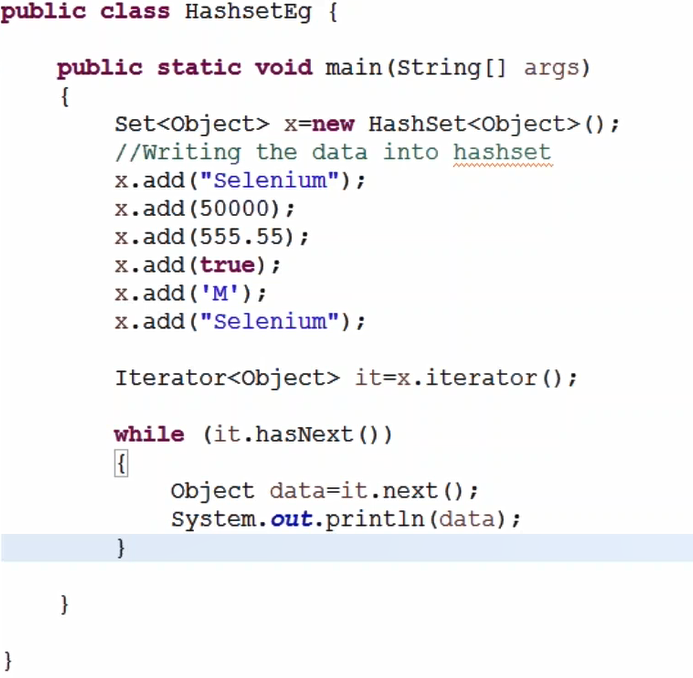
//Traversing list through for-each loop

**for**(String obj:x) {

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

}

**Java HashSet class**



Note: HashSet contains unique elements only(it takes while adding but not print),HashSet allows null value,HashSet doesn't maintain the insertion order