Arrays Strings Vectors, Wrapper Classes

Arrays

Arrays are the data structures which consist of collection of data elements of same data type. Arrays have fixed length once they are declared their size cannot be changed. In JAVA arrays are Objects Arrays can be declared and initialized.

e.g. Creation of an array has three steps

1. Declaring the array
2. Creating memory locations
3. Putting values into memory locations.

Declaration of an array

Syntax

type arrayname[];

type[] arrayname;

e.g.

int no[];

float[] per;

float marks[];

Creation of Arrays

array name = new type[size];

e.g.

no=new int[5];

per=new float[5];

Array Initialization

Type arrayname[ ]={ list of values};

Int no[]={10,20,30,40,50};

Array Length

We can get the length of an array by using arrayname.length.

int var= no.length;

JAVA supports single dimentional,two dimentional ,multidimentional and variable size arrays.

Program to sort an array of integer numbers

```
import java.io.*;
class sort
{
public static void main(String args[])
{
int a[]={14,32,25,17,29};
int i,j,temp;
System.out.println(" elements before sorting are: ");
for(i=0;i<5;i++)
System.out.print("\t"+a[i]);
for(i=0;i<5;i++)
for(j=0;j<4-i;j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
a[j]=a[j+1];
a[j+1]=temp;
}
}
System.out.println(" elements after sorting are: ");
for(i=0;i<5;i++)
System.out.print("\t"+a[i]);
}
}
```

Program to diagonal elements of a 3X3 matrix to zero

```
import java.io.*;
class diagonal
{
public static void main(String args[])
{
int i,j;
int a[][]=new int[3][3];
for(i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
if(i==j)
{
a[i][j]=0;
System.out.print(" "+a[i][j]);
}
else
{
a[i][j]=1;
System.out.print(" "+a[i][j]);
}
}
System.out.print("\n");
}
}
}
```

Strings

A string in JAVA is a sequence of characters.It is also called as array of characters.In JAVA strings are implemented as object of String Class.

How to declare and use string

String stringname;

Stringname= new String(“string”);

e.g.

String firstname;

Firstname= new String(“Pritee”);

How to declare array of strings

String arrayname[]= new String[5];

The above declaration will hold three strings in arrayname.

Most commonly used string functions are as follows

1. s2=s1.toLowerCase() – Converts the string s1 to all lowercase
2. s2=s1.toUpperCase() – Converts the string s1 to all uppercase
3. s2=s1.replace(‘X’,’Y’) – Replace all appearances of X with Y
4. s2=s1.trim() – Remove white spaces at the beginning and end of the string s1
5. s1.equals(s2) – Returns true if s1 is equal to s2
6. s1.equalsIgnoreCase() – Returns true if s1=s2 ignoring the case of characters
7. s1.length() – Gives length of s1
8. s1.CharAt(n)- Gives nth character of s1
9. s1.compareTo(s2) – Returns negative if s1 < s2,positive if s1 > s2 otherwise zero
10. s1.concat(s2) – concatenates s1 and s2
11. s1.substring(n) – Gives substring starting from nth character
12. s1.substring(n,m) – Gives substring starting from nth character upto mth character it does not include mth character
13. StringValueOf(p) – Crates a string object of the parameter p (simple type or object)
14. p.toString() – Creates a string representation of the object p
15. s1.indexOf(‘X’) – Gives the position of the first occurrence of ‘X’ in the string s1
16. s1.indexOf(‘X’,n) – Gives the positon of ‘X’ that occurs after nth position in the string s1
17. String.ValueOf(variable) – Converts the parameter value to string representation

Program to sort an array of names

```
Class Stringsort
{
static String name[] = {“pritee”,”devi”,”dev”,”chins”,”rahul”};
public static void main(String args[])
{
Int size = name.length;
String temp= null;
for(int i=0;i<size;i++)
```

```
{
for(int j=i+1;j<size;j++)
{
if(name[j].compareTo(name[i]) < 0)
{
//swap the strings
temp= name[i];
name[i]= name[j];
name[j]= temp;
}
}
]
for(int i=0;i<size;i++)
{
System.out.println(name[i]);
}
}
}
```

StringBuffer and StringTokenizer

Once a string is assigned a value I canot be changed.It means if a string is reassigned a value the original memory location is lost and new memory location is allotted.To solve this problem JAVA introduced StringBuffer.It is capable of storing a number of characters specified by its capacity.If the capacity of a StringBuffer is exceeded the capacity is automatically expanded to accommodate the additional characters.

Functions used in StringBuffer

1.s1.setCharAt(n,'X) – Modifies the nth character to 'X'

2.s1.append(s2) – Appends the string s2 to s1 at the end

3.s1.insert(n,s2) – Inserts the string s2 at the position n of the string s1

4.s1.setLength(n) – Sets the length of the string s1 to n.If n<s1.length() s1 is truncated.If n>s1.length() zeros are added to s1.

StringTokenizer

A sentence is made up of a number of words called tokens.Compilers also perform tokenization.They breakup the statements into individual pieces like keywords,identifiers,operators and other elements of a programming language.JAVA tokenizer class is used to break the string into its component tokens.Tokens are separated from one another by delimeters,while spaces,blank,tab,new line of carriage return.

Vector

The concept of variable arugments to methods is achieved in JAVA with the use of Vector Class which is present in java.util package.This class is used to create a generic dynamic array known as vector which can hold objects of any type and any number.The objects do not have to be homogenous.Arrays can be easily implemented as vectors.

Vector v = new Vector(); // declaration without size

Vector v=new Vector(3); // declaration with size

Advantages of Vectors over Arrays

1.It is convenient to use vectors to store objects

2.A vector can be used to store a list of objects that may vary in size

3.we can add and delete objects from the list as and when required

A major constraints of vectors is that we cannot directly store simple data type in a vector we can only store objects.

We can convert simple types to objects using wrapper classes.

Important Vector methods are

list.addElement(item) – adds the item to the list at the end

list.elementAt(5) – gives the name of the 5^{th} object

list.size() – gives the number of objects present

list.removeElement(item) – removes the specified item from the list

list.removeElementAt(n) – removes the item stored in the nth position of the list

list.removeAllElements() – removes all the elements in the list

list.copyInto(array) – copies all items from list to array

list.insertElementAt(item,n) – inserts the item at nth position

e.g.

Program to explain working with Vectors

```
import java.util.*;
Class TestVector
{
public static void main(String args[])
{
Vector list= new Vector();
int length=args.length;
for(int i=0;i<length;i++)
{
list.addElement(args[i]);
}
list.insertElementAt("COBOL",2);
int size = list.size();
String listaray[] = new String[size];
list.copyInto(listarray);
System.out.println("List of Languages");
for(int i=0;i<size;i++)
{
System.out.println(listarray[i]);
}
}
}
```

Wrapper Classes

Vectors can not handle primitive data types like int,float,long,double,char etc.Primitive data types can be converted into objects types by using wrapper classes contained in the java.lang package.

Following are the data types and their corresponding wrapper class types.

Simple type	Wrapper class
Boolean	Boolean
Char	Character
Double	Double
Float	Float
Int	Integer
Long	Long

Following are the conversation from primitive data types to object numbers using constructor methods

Ingeger IntVal = new Integer(i) //primitive integer to Integer Object

Converting Object numbers to Primitive numbers using typeValue() method

Int I = IntVal.intValue(); // Object to primitive integer

Converting numbers to Strings using to String() method

Str = Integer.toString(i); //primitive integer to string

Converting String Object to Numeric Objects using the static method ValueOf()

IntVal = Integer.ValueOf(str); // Converts string to Integer Object

Converting Numeric String to Primitive Numbers Using Parsing Methods

int I = Integer.parseInt(str);

e.g

Program how to use wrapper class methods

```
Class test
{
Public static void main(String args[])
{
try
{
DataInputStram in = new DataInputStram(System.in);
System.out.println(" enter a no");
String str= in.readLine();
int a=Integer.parseInt(str);
}
catch(IOException e)
{
System.out.println(" I/O Error");
}
System.out.println("\n the square of a number entered by the user is "+(a*a));
}
}
```