|  |  |
| --- | --- |
|  | Array  1. Array is a collection of similar type of element.  2. We can store only fixed set of element.  3. Array is index based and first element of the array is 0th index    Advantages  1. We can represent huge number of values by using single variable.  2. Random Access  3. Code optimization  Disadvantages :  1. Fixed in size i.e.once we create arrary there is no chance of increasing size.  - TO overcome of this problem we use collection/ArrayList  2. Array  Arrays can hold only similar / homogenous type of data  - To overcome of this problem we use object array.  What happens if we declare an array without assigning the size?  It is not possible to declare an array without size. When we declare an array without assigning the size, it throws the compile-time error. For example, height=new int[].  2) Can we declare array size as negative?  No, the array size cannot be negative. If we declare an array with a negative size, it throws NegativeArraySizeException at run time.  3) When ArrayIndexOutOfBoundsException occurs?  The ArrayIndexOutOfBoundsException occurs when the program tries to access the index of an array. The exception also occurs when the index is higher than the size of the array or the index is negative. |
|  |  |
|  | package Array;  public class FirstProgram {  public static void main(String[] args) {    int i4 = 10;  int i1 =20;  int i3 =30;    // int i10 =10;    // System.out.println(i);  // System.out.println(i1);  // System.out.println(i3);    // Array is a collection of similar type of elements.  // We can store only fixed set of element in java array.  // Array is indexed based and first element of the array is stored at 0th index.    // Array declaration    int arr[] = new int[10];    // Array initialization    arr[0] = 10;  arr[1] = 20;  arr[2] = 30;  arr[3] = 40;  arr[4] = 50;  arr[5] = 60;  arr[6] = 70;  arr[7] = 80;  arr[8] = 90;  arr[9] = 100;  // arr[10] =200; //ArrayIndexOutOfBoundsException    System.out.println(arr.length);  //Array usage air98990  System.out.println(arr[4]);  System.out.println(arr[5]);    System.out.println();  for(int i=0 ;i<arr.length;i++) {  System.out.println(arr[i]);  }  }  } |
|  | package Array;  import java.util.Arrays;  public class Array2 {  public static void main(String[] args) {    char c[] =new char[5]; //- // 0 to 4      c[0] = 'A';  c[1] = 'C';  c[2] = 'z';  c[3] = 's';  c[4] = 'e';  //c[5] = 's';      System.out.println(c.length);    Arrays.sort(c);    for(int k=0;k<c.length;k++) {  System.out.print(" "+c[k]);  }}] |
|  | package Array;  public class Array3 {  public static void main(String[] args) {    double []d = new double[5];      d[0] = 12.22;  d[1] = 22.33;  d[3] = 44.22;  d[2] = 55.33;  d[4] = 44.1;      System.out.println(d[2]);    for(int i=0;i<d.length;i++) {    System.out.println(d[i]);  }  }  } |
|  | package Array;  import java.util.Arrays;  public class Array4 {  public static void main(String[] args) {  // String str [] = new String[-2]; //java.lang.NegativeArraySizeException    String str [] =new String [4];      str[0] = "Rohit";    str[1] = "1234";  str[2] = "!@#$%^&";  str[3] = "Roahit";    System.out.println(str.length);    Arrays.sort(str);  //System.out.println(str);    for(int i=0;i<str.length;i++) {    System.out.println(str[i]);  }    }  } |
|  | package Array;  public class Array5 {  public static void main(String[] args) {      boolean b[] =new boolean[2];    b[0] = true;  b[1] = false;    System.out.println(b[0]);        }  } |
|  | package Array;  public class Array7 {  public static void main(String[] args) {      Object obj [] = new Object[5];    obj[0] = 10;  obj[1] = 12.5f;  obj[2] = "Welcome";  obj[3] = 'S';  obj[4] = true;    System.out.println(obj.length);    for(int i=0 ;i<5 ;i++) {    System.out.println(obj[i]);  }    // Advantages :  //1. We can store multiple element in one variable  //2 .Code optimization  //3 .Random access    //disadvantages    //1. Fixed in size i.e once we create array there is no chance of increasing size.  //( To overcome of this problem we use arrayList)  //2. Array can hold only similar/homogeneous type data.  // To overcome of this problem we use object array  }  } |
|  | package Array;  public class TwoDimensionalarray {  public static void main(String[] args) {  //Declaration    String Str[][] = new String [4][5];      System.out.println(Str.length); // Total no rows    System.out.println(Str[2].length); // Total no columns    // Initialization    //Row & columns    //First row  Str[0][0] = "A1";  Str[0][1] = "A2";  Str[0][2] = "A3";  Str[0][3] = "A4";  Str[0][4] = "A5";    System.out.println(Str[0][3]);    //Second Row  Str[1][0] = "B1";  Str[1][1] = "B2";  Str[1][2] = "B3";  Str[1][3] = "B4";  Str[1][4] = "B5";    //Third Row    Str[2][0] ="C1";  Str[2][1] ="C2";  Str[2][2] ="C3";  Str[2][3] ="C4";  Str[2][4] ="C5";    //4th row  Str[3][0] ="D1";  Str[3][1] ="D2";  Str[3][2] ="D3";  Str[3][3] ="D4";  Str[3][4] ="D5";    // System.out.println(Str[2][1]);    // 0<4 - True  // 1<4 - true  //4<4 - false  for(int row =0 ;row<4;row++) { // Row    // 0<5 - True  // 1<5 -true  //2<5 -true  //5<5 - false  for(int col=0 ;col<5;col++) {    System.out.print(Str[row][col]+" ");  // str[0][0] = A1  // str[0][1] = A2  // str[0][2] = A3  // str[1][0] = B1,B3,B3,B5    }  }    }  } |
|  | package Array;  public class TwoDimensionalArray2 {  public static void main(String[] args) {    // 3\*4  int Number [] [] = {{2,3,4,2},{6,4,3,5},{7,1,3,4}};    System.out.println(Number.length); // Total no of row    System.out.println(Number[1].length);    System.out.println(Number[0][2]);    System.out.println(Number[2][0]);    for(int i=0;i<Number.length;i++) {    for(int j=0;j<Number[0].length;j++) {    System.out.print(Number[i][j]+" ");  }  }    }  } |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |