Vibhor Mehta JAVA DEVOPS

31/05/2018

1)Can abstract class have constructors in Java?

Yes

2) Can abstract class implements interface in Java? do they require to implement all methods?

In Java, an abstract class can implement an interface, and not provide implementations of all of the interface's methods. It is the responsibility of the first concrete class that has that abstract class as an ancestor to implement all of the methods in the interface.

3) Can abstract class be final in Java?

No. A final class is considered complete and cannot be sub classed. A final class can not have abstract methods and an abstract class can not be declared final.

4) Can abstract class have static methods in Java?

Yes it can but you cannot declare static methods in abstract class.

5) Can you create instance of abstract class?

No, you cannot create an instance of an abstract class because it does not have a complete implementation. The purpose of an abstract class is to function as a base for subclasses.

6) Is it necessary for abstract class to have abstract method?

No we can have an abstract class without Abstract Methods as both are independent concepts.

7) Difference between abstract class and interface in Java?

Main difference is methods of a Java interface are implicitly abstract and cannot have implementations. A Java abstract class can have instance methods that implements a default behavior. Variables declared in a Java interface is by default final. An abstract class may contain non-final variables.

8) When do you favor abstract class over interface?

When you have only single inheritance required since java does not support multiple inheritance.   
9) What is abstract method in Java?

An abstract method is a method that is declared without an implementation

10) Can abstract class contains main method in Java ?

Yes it can.  
11) what is static block in java?

Static block is used for initializing the static variables.

12) What is the need of static block?

To help initiate complex initialization in java.

13) Can we overload static methods in java?

Yes we can but we cannot override static methods.

14) Can we call super class static methods from sub class?

Yes we can call super class static method inside sub class using super\_class\_method();

15)What is the difference between final and static keywords?

Once it has been assigned, the value of the final variable cannot change. Static means it belongs to the class not an instance, this means that there is only one copy of that variable/method shared between all instances of a particular Class.

16) Write a note on covariant return type with example code.

Covariant return, means that when one overrides a method, the return type of the overriding method is allowed to be a subtype of the overridden method's return type.

class A {}

class B extends A {}

class Base

{

    A fun()

    {

        System.out.println("Base tennis()");

        return new A();

    }

}

class Derived extends Base

{

    B fun()

    {

        System.out.println("Derived zoom()");

        return new B();

    }

}

public class Main

{

    public static void main(String args[])

    {

       Base base = new Base();

       base.fun();

       Derived derived = new Derived();

       derived.fun();

    }

}

17) Write a note on Enum with example code.

An enum is a special type of data type which is basically a collection of constants. In this tutorial we will learn how to use enums in Java and what are the possible scenarios where we can use them.

public enum Directions{

EAST,

WEST,

NORTH,

SOUTH

}

public class EnumDemo

{

Directions dir;

public EnumDemo(Directions dir) {

this.dir = dir;

}

public void getMyDirection() {

switch (dir) {

case EAST:

System.out.println("In East Direction");

break;

case WEST:

System.out.println("In West Direction");

break;

case NORTH:

System.out.println("In North Direction");

break;

default:

System.out.println("In South Direction");

break;

}

}

public static void main(String[] args) {

EnumDemo obj1 = new EnumDemo(Directions.SOUTH);

obj1.getMyDirection();

EnumDemo obj2 = new EnumDemo(Directions.NORTH);

obj2.getMyDirection();

}

}

18) Write a note on use of super keyword and super() method.

The super keyword in java is a reference variable which is used to refer immediate parent class object.

* super can be used to refer immediate parent class instance variable.
* super can be used to invoke immediate parent class method.
* super() can be used to invoke immediate parent class constructor.

super() calls the parent constructor with no arguments.

19) Write a code to implement abstraction using interface.

interface MyInterface

{

public void method1();

public void method2();

}

class Demo implements MyInterface

{

public void method1()

{

System.out.println("implementation of method1");

}

public void method2()

{

System.out.println("implementation of method2");

}

public static void main(String arg[])

{

MyInterface obj = new Demo();

obj.method1();

}

}

20)Write a Java program to sort a numeric array and a string array.

import java.util.Arrays;

public class Sortnumericandstring {

public static void main(String[] args) {

// TODO Auto-generated method stub

int [] myarray= {24,16,57,9,81};

String [] array1 = {"Car","JAVA","Python", "apple","ball"};

System.out.println("Original numeric array: "+Arrays.toString(myarray));

Arrays.sort(myarray);

System.out.println("Sorted numeric array: "+Arrays.toString(myarray));

System.out.println("Original string array: "+Arrays.toString(array1));

Arrays.sort(array1);

System.out.println("Sorted string array: "+Arrays.toString(array1));

}

21)Write a Java program to sum values of an array.

public class Arraysum {

public static void main(String[] args) {

// TODO Auto-generated method stub

int [] Array = {5,10,15,20,25};

int sum = 0;

for( int count = 0; count<Array.length;count++)

{

sum+=Array[count];

}

System.out.println("SUm of array= "+sum);

}

}  
22)Write a Java program to remove a specific element from an array.

import java.util.Arrays;

public class RemoveArray {

public static void main(String[] args) {

// TODO Auto-generated method stub

int [] array1 = {0,15,13,17,14};

System.out.println("Original array: "+Arrays.toString(array1));

int removeIndex = 1;

for (int i= removeIndex; i<array1.length-1; i++)

{

array1[i] = array1[i+1];

}

System.out.println("After removing element= "+Arrays.toString(array1));

}

}  
23)Write a Java program to reverse an array of integer values.

public class Reversearray {

static void reverseArray(int arr[],int start, int end)

{

int temp;

if (start>end)

return;

temp =arr[start];

arr[start] = arr[end];

arr[end] = temp;

reverseArray(arr,start+1, end-1);

}

static void printArray(int arr[],int size)

{

for(int i=0; i<size; i++)

System.out.print(arr[i]+ " ");

System.out.println("");

}

public static void main(String[] args) {

int arr[] = {1,2,3,4,5,6};

printArray(arr,6);

reverseArray(arr,0,5);

System.out.println("Reversed array is ");

printArray(arr,6);

}

}

24)Write a Java program to find the duplicate values of an array of integer values.

import java.util.Arrays;

public class Exercise12 {

public static void main(String[] args)

{

int[] my\_array = {1, 2, 5, 5, 6, 6, 7, 2};

for (int i = 0; i < my\_array.length-1; i++)

{

for (int j = i+1; j < my\_array.length; j++)

{

if ((my\_array[i] == my\_array[j]) && (i != j))

{

System.out.println("Duplicate Element : "+my\_array[j]);

}

}

}

}

}