Name : MD. Atif Rahman Rudro.

ID : IT-22002

Is it true invoking methods in interface are slower than invoking within abstract classes? Explain and write a new example.

Soln :-

Yes, invoking methods in interface is slower than that of in abstract class due to the following reasons :-

1) Interfaces use interface method tables (itable) that requires more complex runtime lookup compared to virtual tables used by abstract class.

2) A class can implement multiple interfaces, so the JVM needs to resolve which interface method to call. adding slight overhead.

3) Interference methods are not tightly bound to the class hierarchy, so JVM can't optimize them as easily in compile time compared to methods in abstract classes, which has a fixed inheritance path.

Time complexity of methods invoking in both the interference and abstract class is constant time i.e O(1).

Example:

```
interface Animal {
    void makeSound();
}
```

```
abstract class Dog {
        abstract void eat();
    }

class Cow extends Dog implements Animal {
    ~~@Override~~
        public void makeSound () {
            System.out.println(" Cow says Moo! ");
        }

        public void eat() {
            System.out.println("Cow is eating
                                grass!");
        }

}

public class invoking {
        public static void main(String[] args) {
            Animal animal=new Cow();
            long startTime= System.nanoTime();
animal.makeSound();
long endTime1 = System.nanoTime();
System.out.println(" ~~Interference~~ Interface call
      time :"+(endTime1-startTime1)+ "ns");

Dog dog=new Cow();
long startTime2= System.nanoTime();
dog.eat();
long endTime2= System.nanoTime();
System.out.println("Abstract call time : "+
                (endTime2-startTime2)+ "ns");
}
}
```

Output:

```
Cow says Moo!
Interface call time : 419800 ns
Cow is eating grass
Abstract class call time : 56500 ns
```