JAVA

STATIC & FINAL KEYWORD

Static Keyword

The **static keyword** in Java is mostly used for memory management. Static keywords can be used for variables, methods, blocks, and nested classes.

Static Block

- Is used to initialize the static data member.
- It is executed before the main method at the time of classloading.

```
class A2{
   static{System.out.println("static block is invoked");}
   public static void main(String args[])
{
   System.out.println("Hello main");
    }
 }
```

Static Variable

- The static variable can be used to refer to the common property of all objects (which is not unique for each object).
- The static variable gets memory only once in the class area at the time of class loading.

```
public class StaticVariableExample {
// Static variable static
    int a = fun( );
// Static block
    static { System.out.println("Inside Static Block"); }
// Static Method static int fun( ) {
      System.out.println("Inside fun( ) Function");
   return 10;
}
     public static void main(String args[ ]){
        System.out.println("Value of a : "+a);
           System.out.println("Inside Main");
 }
}
```

Static Method

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- A static method can access static data member and can change the value of it.

```
public class StaticMethodExample {
  // Static method
          public static void add(int a, int b) {
          int sum = a + b;
            System.out.println(sum); }
         public static void main(String args[ ]) {
 // Calling static method without creating an object
           add(10, 20);
       }
}
```

Final Keyword

The **final keyword** is only applicable to methods, variables, and classes. It is primarily used to control the user's ability to use **variables, methods, and classes.**

Final variable

- A variable is a final variable if it is declared with the final keyword.
- The final variable's value is assigned only once.
- We cannot modify the value of the final variable after it has been assigned.
- If we attempt to reassign the value, we will get a compile-time error.

```
 public class FinalVariableExample {
 // Final variable.
       final int val = 50;
      void show( ) {
// Trying to change the value of the final variable, 'val'
      val = 100;
     System.out.println(val);
}
      public static void main(String args[ ]) {
      FinalVariableExample obj = new FinalVariableExample( );
     obj.show( );
      }
 }
```

Output:

```
FinalVariableExample.java:9: error: cannot assign a value to
final variable val
val = 100;
 ^
1 error
```

final method

- When a method is declared with the final keyword, it is referred to as the final method.
- The characteristic of the final method is that the child class cannot override it.
- If we do not want our method's implementation to be changed, we should make it final.

```
class FinalMethodExample {
    // Final method
        final void show( ) {
        System.out.println("Inside FinalMethodExample Class
      Method");
    }
}
        public class Rupnath extends FinalMethodExample {
   // Trying to override the final method, 'show'
        void show( ) {
        System.out.println("Inside Rupnath Class Method"); }
        public static void main(String args[ ]) {
       Rupnath obj = new Rupnath( );
       obj.show( );
    }
}
```

Output:

```
Rupnath.java:13: error: show() in CodingNinjas cannot override show() in
FinalMethodExample
void show() {
     ^
overridden method is final
1 error
```

final class

If you make any class as final, you cannot extend it.

```
final class FinalClassExample {

     // Methods and Data Members
 }


   // Trying to extend the final class, 'FinalClassExample'

     public class Rupnath extends FinalClassExample { void
   show( )
 {
       System.out.println("Inside Rupnath Class Method");
 }
        public static void main(String args[ ]) {
       Rupnath obj = new Rupnath( );
       obj.show( );
     }
}
```

Output:

```
Rupnath.java:9: error: cannot inherit from final FinalClassExample
public class CodingNinjas extends FinalClassExample {
                                  ^
1 error
```

Jayesh Deshmukh

Follow

Like Comment Share Save