DAY-46

28 August 2023 15:36

OBJECT CLASSES AND METHODS

- IT IS THE SUPER MOST CLASS IN ENTIRE JAVA HIERACHCHY, i.e, EVERY CLASS IN JAVA IS THE SUB CLASS OF OBJECT CLASS.
- THE OBEJCT CLASS CONTAIN THOSE PROPERTIES WHICH ARE COMMON FOR ALL THE CLASS PRESENT IN JAVA
- THE OBEJCT CLASS IS PRESENT IN JAVA.LANG PACKAGE.
- THE OBEJCT CLASS WILL CONTAIN NO ARGUMENTED CONSTRUCTOR

METHODS IN OBJECT CLASS

- public String toString();
- public int hashcode();
- public boolean equal(Object obj);
- final public class getClass();
- protected Object clone();
- protected void finalize();
- final public void notify();

public String toString()

- THIS METHOD WILL RETURN STRING REPRESENTATION OF AN OBJECT.
- THE STRING REPRESENTATION WILL INCLUDE THE FULLY QUALIFIED NAME OF CLASS AND ALONG WITH A HEXADECIMAL VALUE EQUIVALENT TO HASHCODE VALUE.
    - THE FULLY QUALIFIED NAME WILL BE FOLLOWING FORMAT :
      SYNTAX :

      PACKAGENAME.CLASSNAME@HEXADECIMALVALUE EQUIVALENT TO HASHCODE VALUE

```
package objclass;

public class Test1
{

	public static void main(String[] args)
	{
		Test1 a1 = new Test1();
		System.out.println(a1.toString());

		Test1 a2 = new Test1();
		System.out.println(a2.toString());
	}
```

}

public int hashcode()

- THIS METHOD WILL GENERATE A UNIQUE INTEGER VALUE ASSOCIATED WITH OBJECT.
- THIS METHOD WILL GEENRATE VALUE BASED ON ADDRESS OF OBJECT.
- IF TWO OBJECTS ARE HAVING SAME ADDRESS THEN, HASHCODE WILL BE SAME.

```
package objclass;

public class Test1
{

	public static void main(String[] args)
	{
		Test1 a1 = new Test1();
		System.out.println(a1.hashCode());

		Test1 a2 = new Test1();
		System.out.println(a2.hashCode());

		Test1 a3 = a2;
		System.out.println(a3.hashCode());
	}

}
```

public boolean equal(Object obj);

- THIS METHOD IS USED TO COMPARE CURRENT OBJECT WITH GIVEN OBJECT BASED ON HASHCODE VALUE.
- IF TWO OBJECTS ARE HAVING SAME HASHCODE VALUE THEN THIS METHOD WILL RETURN TRUE ELSE FALSE.

```
package objclass;

public class Test1
{

	public static void main(String[] args)
	{
		Test1 a1 = new Test1();
		Test1 a2 = new Test1();
		System.out.println(a1.equals(a2));

		Test1 a3 = a2;
		System.out.println(a3.equals(a2));
	}

}
```

OVERRIDING OBJECT CLASS METHODS

- WE CAN OVERRIDE METHODS OF OBJECT CLASS

FOR EXAMPLE

```
package objclass;

public class Test1
{
	public String toString()
	{
		return "BitsQ";
	}

	public int hashCode()
	{
		return 711;
	}



	public static void main(String[] args)
	{
		Test1 a1 = new Test1();
		System.out.println(a1.toString());
		System.out.println(a1.hashCode());

		Test1 a2 = new Test1();
		System.out.println(a2.hashCode());
	}

}
```

OVERRDING PUBLIC BOOLEAN EQUALS(OBJECT OBJ)

```
package objclass;

public class Watch
{
	int hour;
	int min;
	int sec;

	public Watch(int hour, int min, int sec)
	{
		this.hour = hour;
		this.min = min;
		this.sec = sec;
	}

	public boolean equals(Object obj)
	{
		 Watch w = (Watch)obj;
```

```
        boolean b = (this.hour==w.hour&&this.min==w.min&&this.sec==w.sec);
        return b;
    }
    public static void main(String[] args)
    {
        Watch w1 = new Watch(15,10,20);
        Watch w2 = new Watch(15,10,20);
        System.out.println(w1.equals(w2));
    }

}
```