DAY-35

03 August 2023 15:35

ACCESS MODIFIER /SPECIFIER

WHAT IS ACCESS MODIFIER ?

- THESE ARE SUED TO SEPCIFY THE VISIBILITY OF MEMBERS OF CLASS
- THE DIFFERENT TYPES OF ACCESS ODIFIERS ARE
    - PUBLIC
    - PRIVATE
    - PROTECTED
    - DEFAULT

PUBLIC

- IF WE DECLARE ANY MEMBER AS PULIC , THEN WE CAN ACCES THE PUBLIC MEMBERS ANYWHERE WITHIN PACKAGE AND OUTSIDE THE PACKAGE
- THEY CAN BE ACCESSED ANYWHERE
- IF THE PULIV MEMBERS IS STATIC THEN WE WILL ACCESS MEMBERS USING CLASS NAME, IF MEMBER IS NON STATIC WE CAN ACCESS MEMBERS USING OBEJCT OF THE CLASS.

```
package demo;

public class Public1
{
	public static int a = 10;
	public double b = 23.45;

}
```

```
package demo;

public class Public2 {

	public static void main(String[] args)
	{
		System.out.println(Public1.a);

		Public1 a1 = new Public1();
		System.out.println(a1.b);

	}

}
```

```
package demo1;

import demo.Public1;

public class Public3 {
```

```
	public static void main(String[] args)
	{
		System.out.println(Public1.a);

		Public1 a1 = new Public1();
		System.out.println(a1.b);


	}

}
```

PRIVATE

- THE PRIVATE MEMBERS CAN BE ACCESSED ONLY WITHIN THE CLASS.
- WE CANNOT ACCESS PRIVATE MEMBERS OF ONE CLASS IN ANOTHER CLASS.
- THE PRIVATE ACCESS MODIFIER PROVIDES HIGHEST SECURITY AND LEAST VISIBILITY.

```
package demo;

public class Private1
{
	private static int a = 123;
	private String s = "BitsQ";
	public static void main(String[] args)
	{
		System.out.println(a);

		Private1 a1 = new Private1();

		System.out.println(a1.s);

	}
}
```

```
package demo;

public class Private2 {

	public static void main(String[] args)
	{
		System.out.println(Private1.a);

		Private1 a1 = new Private1();

		System.out.println(a1.s);

	}

}
```

PROTECTED

- THE PROTECTED MEMBERS OF THE CLASS CAN BE ACCESSED ANYWHERE WITHIN PACKAGE AND OUTSIDE PACKAGE ONLY IF THE CASS IN WHICH WE ARE TRYING TO ACCESS THE PROTECTED MEMBERS SHOULD HAVE **IS-A** RELATIONSHIP WITH THE CLASS TO WHICH PROTECTED MEMBERS BELONG TO.

```
package demo;

public class Protected1
{
	protected static int a  = 33;
	protected float f1 = 45.4f;

	protected void m1()
	{
		System.out.println("inside protected method");
	}

	public static void main(String[] args)
	{
		System.out.println(a);

		Protected1 a1 = new Protected1();
		System.out.println(a1.f1);
	}

}
```

```
package demo;

public class Protected2
{

	public static void main(String[] args)
	{
		System.out.println(Protected1.a);

		Protected1 a1 = new Protected1();
		System.out.println(a1.f1);

	}

}
```

```
package demo1;

import demo.Protected1;

public class Protected3 extends Protected1
{

	public static void main(String[] args)
```

```
	{
		System.out.println(Protected1.a);

		Protected3 a1 = new Protected3();
		System.out.println(a1.f1);
		a1.m1();

	}

}
```

DEFAULT

- THE DEFAULT MEMBERS CAN BE ACCESSED ANYWHERE WITHIN THE PACKAGE AND NOT OUTSIDE THE PACKAGE.

```
package demo;

public class Default1
{
	static int a = 456;
	String s = "BitsQ";

	void m1()
	{
		System.out.println("inside default method");
	}
	public static void main(String[] args)
	{
		Default1 a1 = new Default1();

		System.out.println(a);
		System.out.println(a1.s);
		a1.m1();

	}

}

package demo;

public class Default2 {

	public static void main(String[] args)
	{
		Default1 b1 = new Default1();

		System.out.println(Default1.a);
		System.out.println(b1.s);
		b1.m1();

	}

}
```

```
package demo1;

import demo.Default1;

public class Default3 {

	public static void main(String[] args)
	{
		Default1 b1 = new Default1();

		System.out.println(Default1.a);
		System.out.println(b1.s);
		b1.m1();

	}

}
```