Access Specifiers

Date: 26th Mar 2024

Access specifiers are use to represent scope of members of class(variables,methods,constructor).
In java Access specifiers are classified into 4 types

1. private
2. default
3. protected
4. public

1. private:
If you declare any member of class as private then scope of that member remains only within the class. It can't be access from other classes.

2. default:
If you declare any member of class as default then scope of that member remains only within the package. It can't be access from other packages.
There is no keyword to represent default access specifier.

3. protected:
If you declare any member of class as protected then scope of that member remains only within the package.
that class which is present outside the package can access it by one condition ie. inheritance operation

4. public:
If you declare any member of class as public then scope of that member remains throught the project.

```
package AccessSpecifier1;
```

Example1: Private Access Specifier

```
public class Sample1
{
	//variable
	private int num;

	//constructor
	private Sample1()
	{
		num=10;
	}

	//methods
	private void squareOfNum()
	{
		System.out.println(num*num);
	}

	public static void main(String[] args)
	{
		Sample1 s1=new Sample1();
		System.out.println(s1.num);
		s1.squareOfNum();
	}
}
```

```
package AccessSpecifier1;
public class Sample2
{
	public static void main(String[] args)
	{
		Sample1 s1=new Sample1();
		System.out.println(s1.num);
		s1.squareOfNum();
	}
}
```

Example2: Default Access Specifier

```
package AccessSpecifier1;
public class Sample5
{
	//variable
	int num;		//default access specifier

	//constructor
	Sample5()		//default access specifier
	{
		num=10;
	}

	//constructor
    void squareOfNum()    //default access specifier
	{
		System.out.println(num*num);
	}

    public static void main(String[] args)
    {
	//Accessing default members from same class
		Sample5 s5=new Sample5();
		System.out.println(s5.num);
		s5.squareOfNum();
    }
}
```

```
package AccessSpecifier1;
public class Sample6
{
	public static void main(String[] args)
	{
		//Accessing default members from diff class
		Sample5 s5=new Sample5();
		System.out.println(s5.num);
		s5.squareOfNum();
	}
}
```

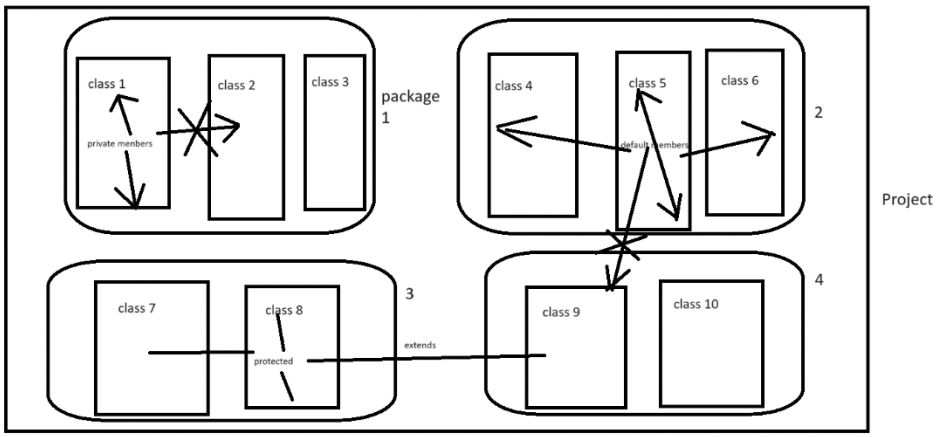
```
package AccessSpecifier2;
import AccessSpecifier1.Sample5;

public class Sample7
{
	public static void main(String[] args)
	{
		Sample5 s5=new Sample5();
		System.out.println(s5.num);
		s5.squareOfNum();
	}
}
```

Example3: Protected Access Specifier

```
package AccessSpecifier1;
public class Sample11
{
	protected int num;  //protected access specifier

	protected Sample11()    //protected access specifier
	{
		num=10;
	}

	protected void squareOfNum()  //protected access specifier
	{
		System.out.println(num*num);
	}


	public static void main(String[] args)
	{
		//Accessing protected from current class
		Sample11 s11=new Sample11();
		System.out.println(s11.num);
		s11.squareOfNum();
	}
}
```

```
package AccessSpecifier1;
public class Sample12
{
	public static void main(String[] args)
	{
		//Accessing protected from diff class
		Sample11 s11=new Sample11();
		System.out.println(s11.num);
		s11.squareOfNum();
	}
}
```

```
package AccessSpecifier2;
import AccessSpecifier1.Sample11;

public class Sample13 extends Sample11
{
	public static void main(String[] args)
	{
		//Accessing protected from diff package without inheritance
//		Sample11 s11=new Sample11();
//		System.out.println(s11.num);
//		s11.squareOfNum();

		Sample13 s13=new Sample13();
		System.out.println(s13.num);
		s13.squareOfNum();
	}
}
```



3: Example of Public Access Specifier

```
package AccessSpecifier1;
public class Sample21
{
	public int num;


	public Sample21()
	{
		num=10;
	}

	public void squareOfNum()
	{
		System.out.println(num*num);
	}

	public static void main(String[] args)
	{
		Sample21 s21=new Sample21();
		System.out.println(s21.num);
		s21.squareOfNum();
	}
}
```

```
package AccessSpecifier1;
public class Sample22
{
	public static void main(String[] args)
	{
		Sample21 s21=new Sample21();
		System.out.println(s21.num);
		s21.squareOfNum();
	}
}
```

```java
package AccessSpecifier2;
import AccessSpecifier1.Sample21;

public class Sample23
{
	public static void main(String[] args)
	{
		Sample21 s21=new Sample21();
		System.out.println(s21.num);
		s21.squareOfNum();
	}
}
```

Creating multiple classes in same class file:-

```java
package AccessSpecifier1;
public class Demo1
{
	public static void main(String[] args)
	{
		m1();
		Demo2.m2();
		Demo3.m3();

		Demo3 d3=new Demo3();
		d3.m4();
	}

	public static void m1()
	{
		System.out.println("running m1 method from Demo1 class");
	}
}


class Demo2	//default access specifier
{
	public static void m2()
	{
		System.out.println("running m2 method from Demo2 class");
	}
}


class Demo3	//default access specifier
{
	public static void m3()
	{
		System.out.println("running m3 method from Demo3 class");
	}

	public  void m4()
	{
		System.out.println("running m4 method from Demo3 class");
	}
}
```

1. What are Access modifiers in java?
2. Why are access modifiers used?
3. How many types of modifiers in Java?
4. Which access modifier is also known as Universal access modifier?
-> Public
5. Which is the most restrictive access modifier in Java?
-> Private
6. Which is the least restrictive access modifier in Java?
-> Public
7. Can we have a private constructor in Java?
8. What is the role of private constructor in Java?
9. Which access modifiers can be used with a class?
-> Public and Default.
7. Can we declare a class as private?
-> No
9. Can we declare a class as protected?
-> No

10. Types of modifiers in java?
1: access modifiers
2: non-access modifiers

11. What are non-access modifiers in Java?
-> There are four non-access modifiers in Java. They are as follows:

1: static
2: final
3: abstract
4: synchronized