

In Java, difference between default, public, protected, and private

Access Modifiers in Java.

Java access modifiers are used to provide access control in java.

1. Default:

Accessible to the classes in the same package only.

e.g.

```
//save by A.java
package pack;
class A{
    void msg(){System.out.println("Hello");}
}
```

```
//save by B.java
package mypack;
import pack.*;
class B{
    public static void main(String args[]){
        A obj = new A();//Compile Time Error
        obj.msg();//Compile Time Error
    }
}
```

This access is more restricted than public and protected but less restricted than private.

2. Public

Can be accessed from anywhere. (Global Access)

e.g.

```
//save by A.java
```

```
package pack;  
public class A{  
    public void msg(){System.out.println("Hello");}  
}
```

```
//save by B.java
```

```
package mypack;  
import pack.*;  
  
class B{  
    public static void main(String args[]){  
        A obj = new A();  
        obj.msg();  
    }  
}
```

Output:Hello

3. Private

Accessible only inside the same class.

e.g.

If you try to access private members on one class in another will throw compile error. e.g.

```
class A{  
    private int data=40;  
    private void msg(){System.out.println("Hello java");}  
}  
  
public class Simple{  
    public static void main(String args[]){  
        A obj=new A();  
        System.out.println(obj.data);//Compile Time Error  
        obj.msg();//Compile Time Error  
    }  
}
```

4. Protected

Accessible only to the classes in the same package and to the subclasses

e.g.

```
//save by A.java
package pack;
public class A{
protected void msg(){System.out.println("Hello");}
}

//save by B.java
package mypack;
import pack.*;

class B extends A{
    public static void main(String args[]){
        B obj = new B();
        obj.msg();
    }
}
```

Output:Hello

| Most Restrictive ← → Least Restrictive | | | | |
|--|---------|-------------------|-----------|--------|
| Access Modifiers -> | private | Default/no-access | protected | public |
| Inside class | Y | Y | Y | Y |
| Same Package Class | N | Y | Y | Y |
| Same Package Sub-Class | N | Y | Y | Y |
| Other Package Class | N | N | N | Y |
| Other Package Sub-Class | N | N | Y | Y |

Same rules apply for inner classes too, they are also treated as outer class properties

answered Aug 7 at 14:16

[share](#) | [edit](#) | [flag](#) edited Aug 7 at 14:24

[Affy](#)
1159