ACCESS MODIFIERS IN JAVA

Velocity

JULY 27, 2022

VELOCITY

Pune

Access Modifiers in java

Access Specifiers-

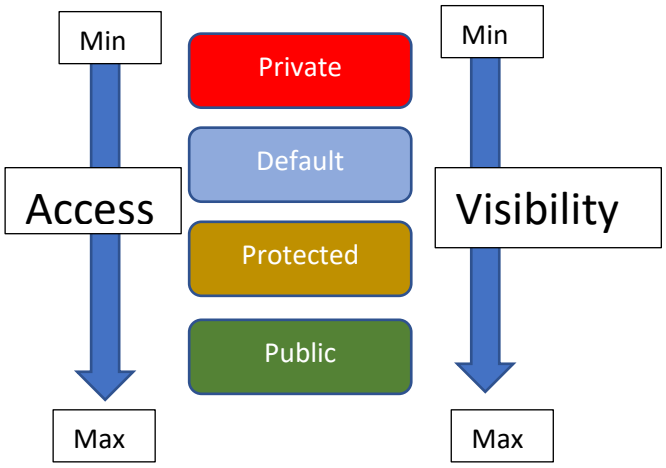
Access modifiers are keywords that can be used to control the visibility of fields, methods, constructors and class. In other term, it is used to restrict the access.

There are four types of access specifiers as

- Private
- Default
- Protected
- Public

Access Modifier	Details
Private	We can access the private modifier only within the same class and not from outside the class.
Default:	We can access the default modifier only within the same package and not from outside the package. And also, if we do not specify any access modifier it will automatically consider it as default.
Protected:	We can access the protected modifier within the same package and also from outside the package with the help of the child class. If we do not make the child class, we cannot access it from outside the package. So, inheritance is a must for accessing it from outside the package.
Public:	We can access the public modifier from anywhere. We can access public modifiers from within the class as well as from outside the class and also within the package and outside the package.

Access Modifier	Within the class	Within the same package	Subclass	In another package
private	Access Allowed	Access Denied	Access Denied	Access Denied
default	Access Allowed	Access Allowed	Access Denied	Access Denied
protected	Access Allowed	Access Allowed	Access Allowed	Access Denied
public	Access Allowed	Access Allowed	Access Allowed	Access Allowed

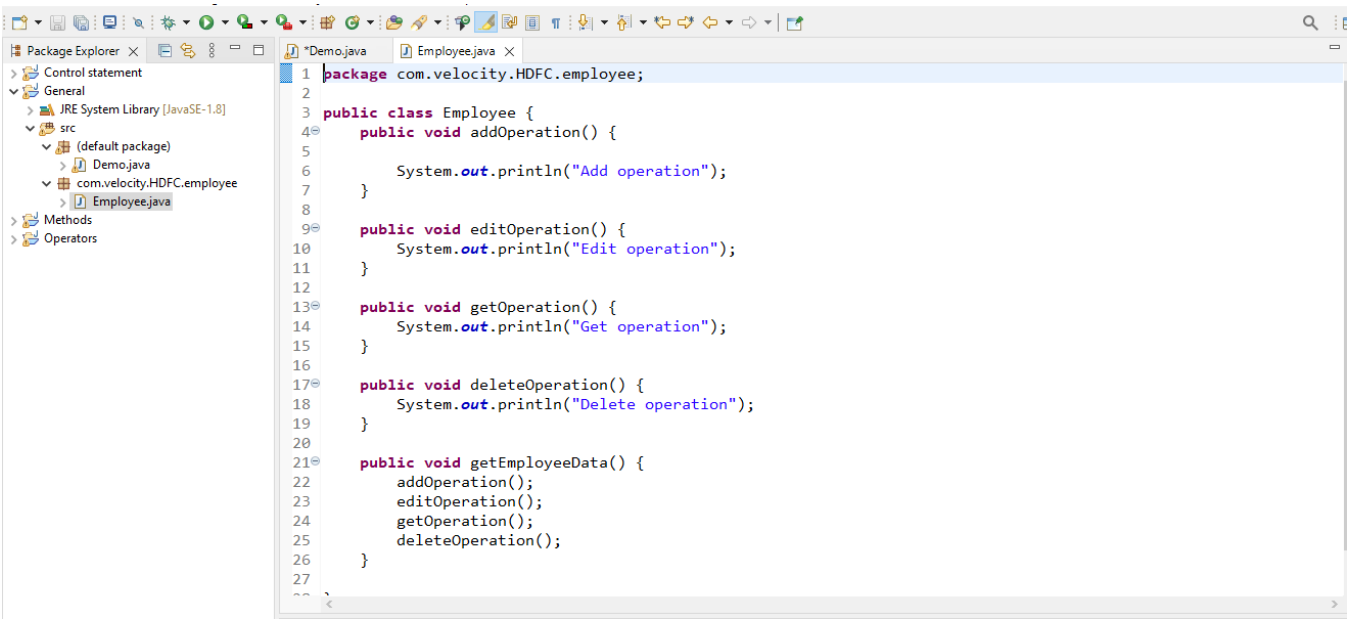


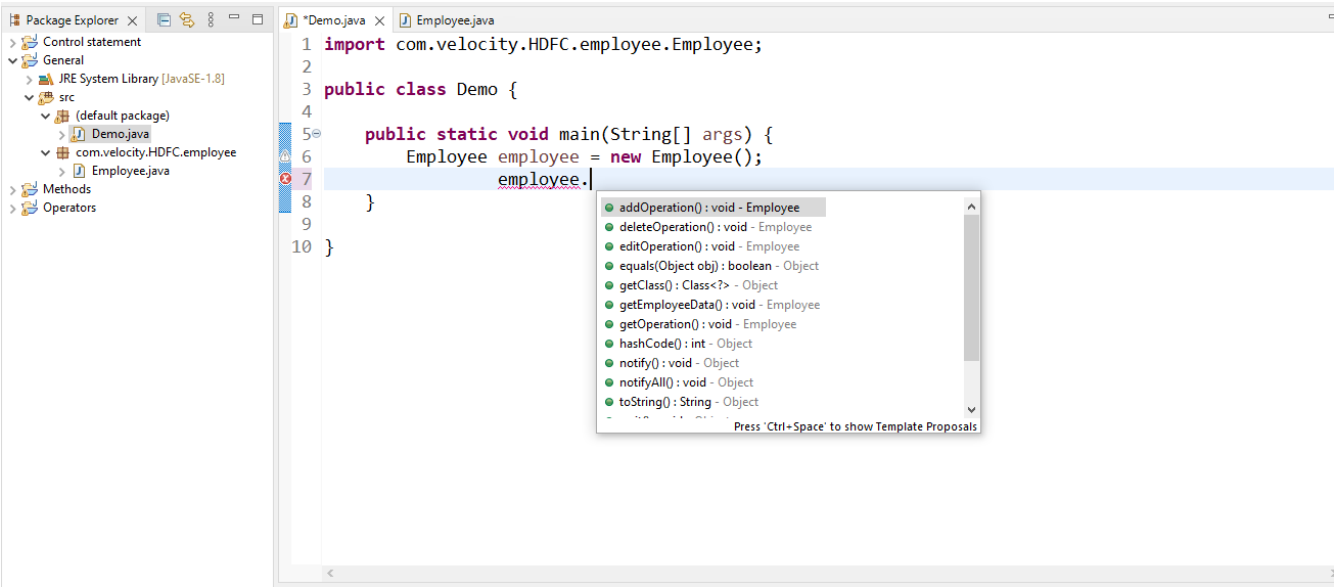
Let us see which all members of Java can be assigned with the access modifiers:

Java Members	Private	default	Protected	public
Class (Outer)	NO	YES	NO	YES
Inner class	YES	YES	YES	YES
Method	YES	YES	YES	YES
Constructor	YES	YES	YES	YES
Global variable	YES	YES	YES	YES
Local variable	NO	YES	NO	NO
interface	NO	YES	NO	YES

Why we use access specifiers?

If we have a business requirement where we need to perform the employee CRUD operations and all the methods need to be called from getEmployeeData() only.





Here we are directly call any method from outside class because scope is public. Hence requirement is not fulfilled here.

```java
package com.velocity.HDFC.employee;

public class Employee {
    private void addOperation() {

        System.out.println("Add operation");
    }

    private void editOperation() {
        System.out.println("Edit operation");
    }

    private void getOperation() {
        System.out.println("Get operation");
    }

    private void deleteOperation() {
        System.out.println("Delete operation");
    }

    public void getEmployeeData() {
        addOperation();
        editOperation();
        getOperation();
        deleteOperation();
    }
```

```java
import com.velocity.HDFC.employee.Employee;

public class Demo {

    public static void main(String[] args) {
        Employee employee = new Employee();
                employee.em
    }

}
```

getEmployeeData() : void - Employee

Press 'Ctrl+Space' to show Template Proposals

```java
import com.velocity.HDFC.employee.Employee;

public class Demo {

    public static void main(String[] args) {
        Employee employee = new Employee();
                employee.getEmployeeData();
    }

}
```

```
Add operation
Edit operation
Get operation
Delete operation
```

Here we cannot directly call any method except getEmployeeData() because scope is private. So, it cannot be directly accessible from outside. We need to access it from by calling getEmployeeData ().

In this way, we use the access specifiers.