

Leksioni 6

Package/ Access Package

6.1 Access Modifiers in Java

Ekzistojnë dy lloje të modifikuesve në Java: modifikuesit e aksesit dhe modifikuesit jo të aksesit. Modifikuesit e hyrjes në Java specifikojnë mundësinë e hyrjes ose qëllimin e një fushe, metode, konstruktori ose klase. Ne mund të ndryshojmë nivelin e qasjes së fushave, konstruktorëve, metodave dhe klasës duke aplikuar modifikuesin e qasjes në të. Ekzistojnë katër lloje të modifikuesve të aksesit në Java:

1. **Private:** Niveli i hyrjes së një modifikuesi privat është vetëm brenda klasës. Nuk mund të arrihet nga jashtë klasës.
2. **Default:** Niveli i hyrjes së një modifikuesi të paracaktuar është vetëm brenda paketës. Nuk mund të arrihet nga jashtë paketës. Nëse nuk specifikoni ndonjë nivel të hyrjes, do të jetë e paracaktuar.
3. **Protected:** Niveli i qasjes së një modifikuesi të mbrojtur është brenda paketës dhe jashtë paketës përmes klasave të derivuara. Nëse nuk ndërtohen klasa të derivuara, nuk mund të arrihet nga jashtë paketës.
4. **Publik:** Niveli i leximit të një modifikuesi publik është kudo. Mund të arrihet nga brenda klasës, jashtë klasës, brenda paketës dhe jashtë paketës.

Ka shumë modifikues pa hyrje, të tillë si statik, abstrakt, i sinkronizuar, vendas, i paqëndrueshëm, kalimtar, etj. Këtu, ne do të mësojmë vetëm modifikuesit e hyrjes të përmendur më lart.

6.2 Analizimi i qasjes së të dhënave të modifikuara në java.

Le të kuptojmë modifikuesit e hyrjes në Java nga një tabelë e thjeshtë.

Access Modifier	Within class	Within package	Outside package by subclass only	Outside package
Private	Yes	No	No	No
Default	Yes	Yes	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

6.2.1 Modifikuesi privat i qasjes

Modifikuesi privat i qasjes është i arritshëm vetëm brenda klasës. Shembull i thjeshtë i modifikuesit të aksesit privat. Në këtë shembull, ne kemi krijuar dy klasa A dhe Simple. Një klasë përmban anëtarin e të dhënave private dhe metodën private. Ne po i aksesojme këto anëtarë privatë nga jashtë klasës, kështu që ka një gabim në momentin e përpilimit.

```
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
}
```

Roli i Konstruktorit Privat

Nëse nuk krijoni ndonjë funksion për të dhënat private të klasës nuk mund të lexoni këto të dhëna të klasës. Për shembull:

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

6.2.2 Default

Nëse nuk përdorni ndonjë modifikues, ai trajtohet si *default by default*. Modifikuesi i paracaktuar është i arritshëm vetëm brenda paketës, nuk mund të arrihet nga jashtë saj. Ai siguron më shumë akses se sa tek të dhënat private. Por, është më kufizuese sesa të dhënat e mbrojtura dhe publike. Shembull i modifikuesit per të dhëna default.

Në këtë shembull, ne kemi krijuar dy paketa pack dhe mypack. Ne jemi duke hyrë në klasën A nga jashtë paketës së saj, duke qënë se kjo klasë nuk është publike, atëherë nuk mund të aksesohet nga jashtë paketës.

```
//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
    }
}
```

Në shembullin e mësipërm, fusha e klasës A dhe metoda e saj msg () është e paracaktuar kështu që nuk mund të lexohet nga jashtë paketës.

6.2.3 Të mbrojtura

Modifikuesi i qasjes së mbrojtur është i arritshëm brenda paketës dhe jashtë paketës, por vetëm përmes trashëgimisë. Modifikuesi i të dhënave të mbrojtur mund të zbatohet në anëtarin e të dhënave duke përdorur metodën dhe konstruktoren. Ndryshe nuk mund të zbatohet në klasë.

Ai siguron më shumë qasje sesa modifikuesi i paracaktuar.

Shembull i modifikuesit të qasjes së mbrojtur:

Në këtë shembull, ne kemi krijuar dy paketat pack dhe mypack. Një klasë e pack tek paketa është publike, kështu që mund të aksesohet nga jashtë paketës. Por metoda msg e kësaj pakete deklarohet e mbrojtur, kështu që mund të arrihet nga jashtë klasës vetëm përmes trashëgimisë.

```
//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

6.2.4 Modifikuesit e aksesit publik

Modifikuesit publik janë më të zakonshmit në aplikacionet Java. Ato nënkuptojnë që çdo klasë, metodë, variabël, ndërfaqe ose konstruktor publik është i arritshëm në të gjithë paketën. Para se të përdorni një klasë publike në një paketë tjetër, së pari duhet ta importoni. Nëse trashëgoni një klasë publike, atëherë nënklasa do të ketë të gjitha metodat dhe ndryshoret e saj si parazgjedhje. Për informacionin tuaj, metoda "main ()" në Java është publike.

```
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

6.3 Modifikuesit e aksesimit në Java nëpërmjet metodse Overriding.

Nëse po harroni ndonjë metodë, metoda e mbivendosur (d.m.th. e deklaruar në nënklasë) nuk duhet të jetë më kufizuese përkundrazi mund të përdoret më së miri.

```
class A{
protected void msg(){System.out.println("Hello java");}
}

public class Simple extends A{
```

```
void msg(){System.out.println("Hello java");} //C.T.Error
public static void main(String args[]){
    Simple obj=new Simple();
    obj.msg();
}
```

Modifikuesi i paracaktuar (default) është më kufizues sesa i mbrojtur(protected) . Kjo është arsyeja pse, ekziston një gabim në kohën e përpilimit.

6.5 Modifikuesit Jo-Aksesues

Në programimin Java, ju do të merrni shtatë modifikues pa qasje. Ju mund t'i implementoni me klasa / metoda / ndryshore / konstruktorë dhe t'i përdorni ato në programe. Këtu janë disa pika për t'ju ndihmuar të kuptoni se kur duhet të përdorni një modifikues të veçantë.

Statike - Do të thotë që diçka lidhet drejtpërdrejt me një klasë. Modifikuesi statik përdoret për të bërë fushën statike. Ne mund ta përdorim atë për të deklaruar ndryshoren statike, metodën, klasën etj. Statike mund të përdoret për të deklaruar variablin e nivelit të klasës. Nëse një metodë deklarohet statike, atëherë nuk kemi nevojë të kemi objekt për ta përdorur atë. Ne mund të përdorim statike për të krijuar një klasë të mbivendosur.

Përfundimtar - Tregon që objekti është i pandryshueshëm.

Abstrakt - Do të thotë që duhet të nënsklasifikoni për krijimin e objekteve.

Sinkronizuar - Kjo tregon se vetëm një thread mund të ekzekutojë një metodë në të njëjtën kohë. Kur një metodë sinkronizohet, ajo mund të arrihet nga vetëm një fije në të njëjtën kohë. Ne do ta diskutojmë atë në detaje në Thread.

Kalimtare - Do të thotë të përjashtosh diçka gjatë shkrimit ndër rreshta. Kur një ndryshore e instancës deklarohet si kalimtare, atëherë vlera e saj nuk vazhdon kur një objekt serializohet. Modifikuesi i paqëndrueshëm/kalimtar i tregon përpiluesit se ndryshorja e paqëndrueshme mund të ndryshohet papritur nga pjesët e tjera të një programi. Variablat e paqëndrueshëm përdoren në rast të programit multi-thread. fjala kyçe e paqëndrueshme nuk mund të përdoret me një metodë ose një klasë. Mund të përdoret vetëm me një ndryshore.

I paqëndrueshëm - Kjo tregon që threade të ndryshme mund të modifikojnë vlerën e një ndryshoreje.

Gjuha amtare - Kjo tregon se metoda është në dispozicion në kodin vendas duke përdorur JNI ose JNA.

6.6 Pikat kryesore

Këtu është një përmbledhje e shpejtë e koncepteve të mësipërme. Kjo do t'ju ndihmojë të përdorni më mirë modifikuesit e qasjes në Java.

Nëse po shkruani një klasë për ripërdorim, atëherë aplikoni nivelin më kufizues të hyrjes në ato fusha ku është logjike.

Gjithashtu, bëjeni aksesin privat zgjedhjen tuaj të parazgjedhur derisa të ketë një rast të kundërt. Për më tepër, mos etiketoni kurrë identifikuesit e vazhdueshëm si publik.

Modifikuesi pa qasje

Së bashku me modifikuesit e aksesit, Java gjithashtu ofron modifikues pa qasje. Këto modifikues përdoren për të vendosur veti të veçanta në variabël ose metodë. Modifikuesit pa qasje nuk e ndryshojnë mundësinë e hyrjes së ndryshores ose metodës, por ato ofrojnë veti të veçanta për ta. Java ofron modifikuesit e mëposhtëm jo-aksesues. Modifikuesi përfundimtar mund të përdoret me variabël, metodë dhe klasë. Nëse ndryshorja deklarohet përfundimtare atëherë nuk mund ta ndryshojmë vlerën e saj. Nëse metoda deklarohet përfundimtare, atëherë ajo nuk mund të anashkalohet dhe nëse një klasë deklarohet përfundimtare, atëherë nuk mund ta trashëgojmë atë.