

Tema: Java Access Modifiers

Tabela e përmbajtjes

Lista e figurave	3
Çfarë janë Java modifiers?	4
1. Public Access Modifier	4
2. Protected Access Modifier	6
3. Default Access Modifier	9
4. Private Access Modifier	11
Përfundim	13

Lista e figurave

Fig. 1 Klasa Shpend (Public Access Modifier)	5
Fig. 2 Klasa TestClass (Public Access Modifier)	5
Fig. 3 Output TestClass (Public Access Modifier)	5
Fig. 4 Klasa TestClassForPublic (Public Access Modifier)	6
Fig. 5 Output TestClassForPublic (Public Access Modifier)	6
Fig. 6 Klasa Shpend (Protected Access Modifier)	7
Fig. 7 Klasa TestClassProtected (Protected Access Modifier)	7
Fig. 8 Output TestClassProtected (Protected Access Modifier)	7
Fig. 9 Klasa HarabelForProtected (Protected Access Modifier)	8
Fig. 10 Klasa HarabelForProtected pa trashegimi (Protected Access Modifier).....	8
Fig. 11 Klasa Shpend (Default Access Modifier)	9
Fig. 12 Klasa TestClassDefault (Default Access Modifier)	9
Fig. 13 Output TestClassDefault (Default Access Modifier)	10
Fig. 14 Klasa TestClassForDefault (Default Access Modifier).....	10
Fig. 15 Klasa TestClassForDefault me trashegimi (Default Access Modifier)	10
Fig. 16 Klasa Shpend pa Getters & Setters (Private Access Modifier)	11
Fig. 17 Klasa TestClassPrivate (Private Access Modifier)	11
Fig. 18 Klasa Shpend me Getters & Setters (Private Access Modifier)	12
Fig. 19 Klasa TestClassPrivate duke perdorur Getters & Setters (Private Access Modifier)	13
Fig. 20 Output TestClassPrivate duke perdorur Getters & Setters (Private Access Modifier)	13

Çfarë janë Java modifiers?

Modifiers në Java janë fjalë kyçe (*keywords*) të cilat ia shton variablave, metodave, konstruktorëve ose klasave për t'i ndryshuar kuptimin.

Java ka dy grupe modifiers:

- access modifiers
- non-access modifiers

Në këtë artikull do të flasim për **Access Modifiers**.

Çfarë janë Access Modifiers? Nga vet emri *Access Modifiers* (modifikuesit e aksesit) ata specifikojnë aksesibilitetin apo scope (fushën e veprimit) të një variabli, metode, konstruktori apo klase.

Tipet e Access Modifiers janë si më poshtë:

- Public
- Protected
- Default
- Private

Le t'i shohim të gjitha me radhë dhe t'i kuptojmë më mirë me shembuj. Si fillim shpjegojmë çfarë është një **package** në Java. Një *package* është një hapësirë që organizon një grup klasash dhe ndërfaqesh (interfaces) të ndërlidhura. Konceptualisht mund ta mendojmë si *folders* që krijojmë në kompjuterin tonë. Në këtë rast klasat / ndërfaqet mund t'i mendojmë si file-t të cilat gjenden brenda këtij folder. Përdorim *packages* për të shmangur konfliktet e emrave dhe për të shkruar një kod më të mirë/pastër.

1. Public Access Modifier

Kur metodat, variablat, klasat etj. deklarohen publike, atëherë mund t'i aksesojmë nga **kudo**. *Public Access Modifier* nuk ka kufizime në fushën e veprimit. Mund të aksesohet nga brenda klasës, jashtë klasës, brenda *package* dhe jashtë saj. Shohim kodin e mëposhtëm. Mora përsëri shembull me Shpendët meqë jeni familjarizuar. Në *package* **PublicAccessModifier** kemi dy klasat publike **Shpend** dhe **TestClass**. Në klasën **Shpend** kemi deklaruar *variablin public të instancës* *cicërima* të tipit String, si dhe metodën publike **shfaq()**. Kjo metodë bën të mundur printimin e tekstit të vendosur në thonjëza. Printimi i tekstit të dytë printon vlerën *cicërima*, e cila vendoset në klasën **TestClass**. Në klasën **TestClass** shohim në fillim rreshtin 1, **package PublicAccessModifier** e cila tregon se kjo klasë ndodhet brenda këtij *package*. Më pas brenda metodës **main()** kemi krijuar objektin e tipit **Shpend**. Në rreshtin 10 aksesojmë variablin publik të klasës **Shpend** dhe i japim vlerën "Ciu ciu". Më pas thërrasim metodën publike **shfaq()**. Në output shohim se pintoen dy fjalitë.

```

1  package PublicAccessModifier;
2
3  //Klase publike
4  public class Shpend {
5      // variabel instance publike
6      public String cicerima;
7
8      // metode publike
9      public void shfaq() {
10         System.out.println("Shpendët cicerijne");
11         System.out.println("Cicerima e shpendeve tingellon " + cicerima);
12     }
13 }

```

Fig. 1 Klasa Shpend (Public Access Modifier)

```

1  package PublicAccessModifier;
2
3  // Klase publike
4  public class TestClass {
5      public static void main(String[] args) {
6          // Krijojme objektin Shpend duke aksesuar klasen publike
7          Shpend shpend = new Shpend();
8
9          // Aksesojme variablin publik
10         shpend.cicerima = "Ciu ciu";
11
12         // Aksesojme metoden publike shfaq()
13         shpend.shfaq();
14     }
15 }

```

Fig. 2 Klasa TestClass (Public Access Modifier)

```

Shpendët cicerijne
Cicerima e shpendeve tingellon Ciu ciu

```

Fig. 3 Output TestClass (Public Access Modifier)

Kemi krijuar gjithashtu një *package* tjetër me emrin **OtherPackageForAccessModifiers**. Kështu mund të kuptoni më mirë aksesibilitetin e secilës klasë. Kemi krijuar klasën **TestClassForPublic** ku brenda metodës `main()` kodojmë të njëjtën gjë si në klasën `TestClass`. Duke qenë se klasa `Shpend` dhe variabli / metoda brenda saj janë publike atëherë mundemi t'i aksesojmë dhe në këtë klasë që ndodhet në një *package* tjetër. Output-i është i njëjtë me klasën `TestClass`.

```
TestClassForPublic.java x
1 package OtherPackageForAccessModifiers;
2
3 import PublicAccessModifier.Shpend;
4
5 public class TestClassForPublic {
6     public static void main(String[] args) {
7         Shpend shpend = new Shpend();
8         shpend.cicerima = "Ciu ciu";
9         shpend.shfaq();
10    }
11 }
```

Fig. 4 Klasa `TestClassForPublic` (Public Access Modifier)

```
Shpendët cicerijne
Cicerima e shpendeve tingellon Ciu ciu
```

Fig. 5 Output `TestClassForPublic` (Public Access Modifier)

2. Protected Access Modifier

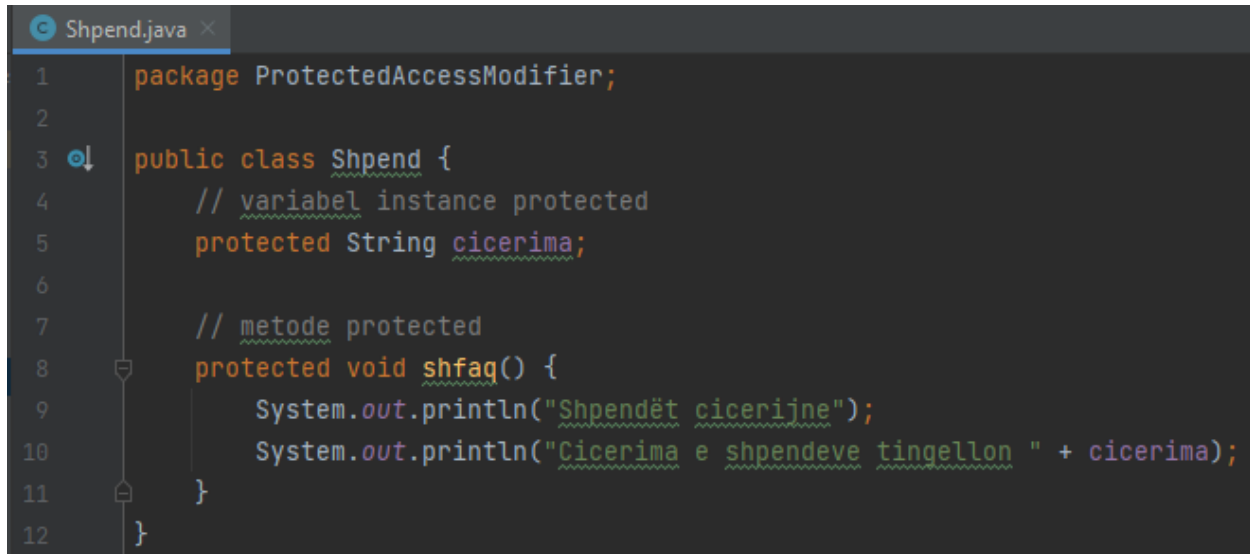
Kur metodat ose variablat e një klase deklarohen **protected**, mund t'i aksesojmë brenda të njëjtës *package* si dhe nga *subclasses*. Një klasë në Java *nuk deklarohet si protected*, vetëm pjesëtarët e saj.

Pse një klasë nuk deklarohet *protected* në Java?

Sepse *protected* përdoret për aksesim brenda të njëjtës *package* si dhe nga *subclasses* që trashëgojnë (inherit) klasën (edhe në *packages* të tjera). Nëse klasa do ishte *protected*, atëherë klasa të tjera jashtë *package* ku ndodhet nuk do të mund ta trashëgonin dot dhe do të ishte njëlloj sikur të përdorim **Default**.

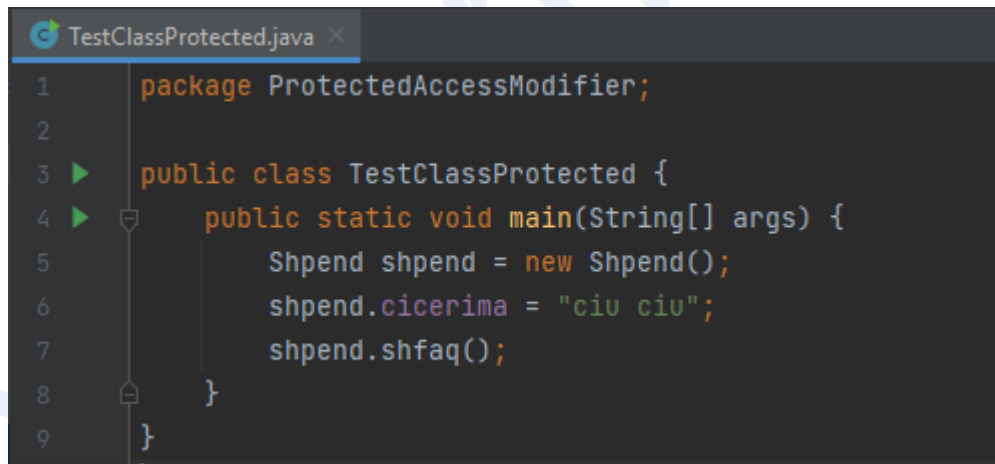


Shohim shembullin e mëposhtëm, për ta kuptuar më qartë. Në package **ProtectedAccessModifier** krijojmë klasën **Shpend** si më sipër, por në vend të *public* i deklarojmë pjesëtarët e klasës *protected*. Në klasën **TestClassProtected** krijojmë objektin e tipit Shpend dhe veprojmë si më parë, duke i dhënë vlerë variablit cicerima dhe thërrasim metodën shfaq().



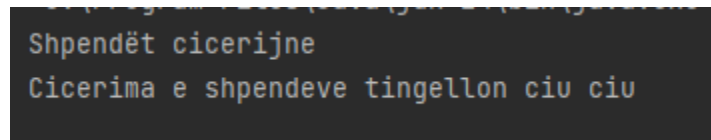
```
1 package ProtectedAccessModifier;
2
3 public class Shpend {
4     // variabel instance protected
5     protected String cicerima;
6
7     // metode protected
8     protected void shfaq() {
9         System.out.println("Shpendët cicerijne");
10        System.out.println("Cicerima e shpendeve tingellon " + cicerima);
11    }
12 }
```

Fig. 6 Klasa Shpend (Protected Access Modifier)



```
1 package ProtectedAccessModifier;
2
3 public class TestClassProtected {
4     public static void main(String[] args) {
5         Shpend shpend = new Shpend();
6         shpend.cicerima = "ciu ciu";
7         shpend.shfaq();
8     }
9 }
```

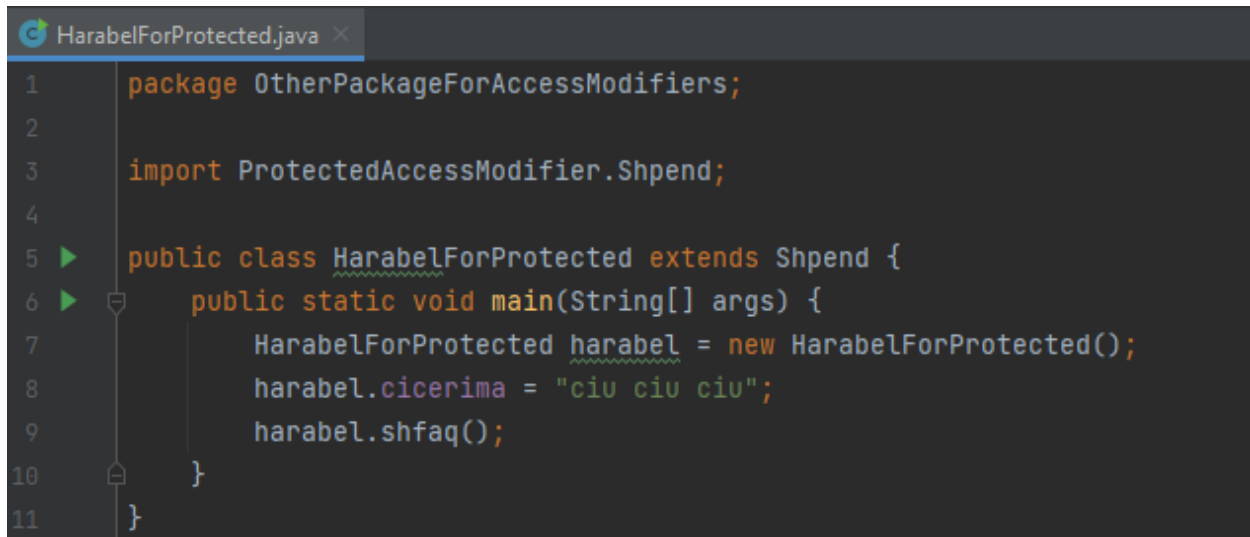
Fig. 7 Klasa TestClassProtected (Protected Access Modifier)



```
Shpendët cicerijne
Cicerima e shpendeve tingellon ciu ciu
```

Fig. 8 Output TestClassProtected (Protected Access Modifier)

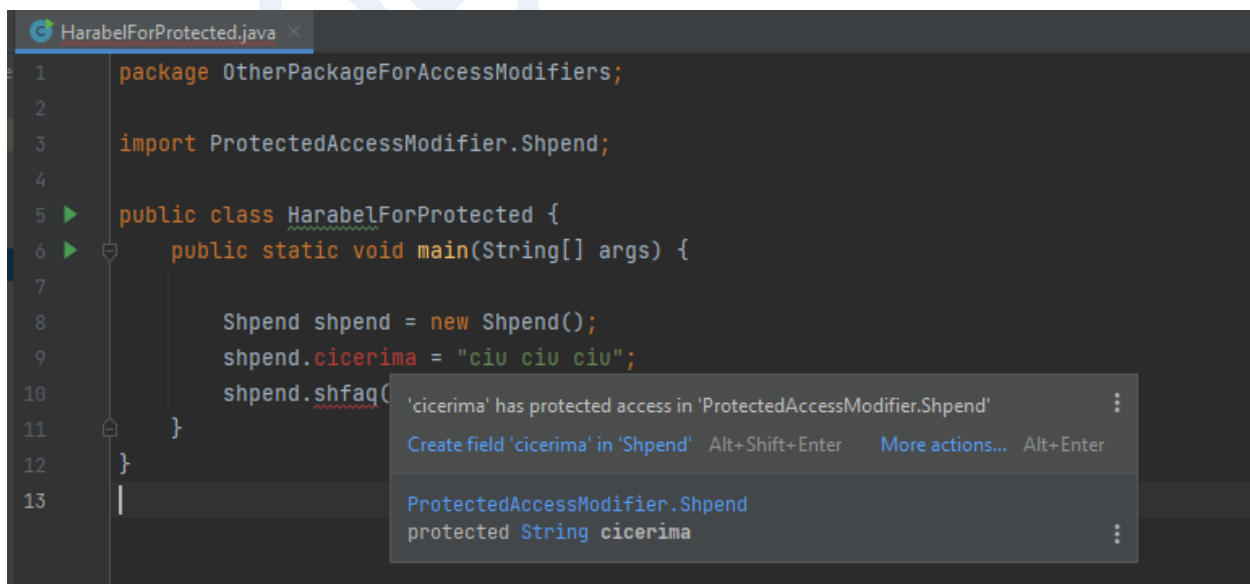
Deri tani nuk shohim ndonjë ndryshim nga *public* apo jo? Në *package OtherPackageForAccessModifiers* krijojmë klasën **HarabelForProtected** (mesa duket *@programuesja* njih vetëm një lloj shpendi). Nga kjo klasë trashëgojmë klasën *Shpend* dhe më pas krijojmë objektin e tipit *Harabel*. Nga ky objekt i japim vlerë variablit *cicerima* dhe thërrasim metodën *shfaq()*. (Trashëgimia/Inheritance është shpjeguar në [artikullin e OOP](#)). Nëse nuk do e trashëgonim klasën *Shpend* atëherë nuk do të mund të përdornim dot pjesëtarët e saj të klasës.



```
1 package OtherPackageForAccessModifiers;
2
3 import ProtectedAccessModifier.Shpend;
4
5 public class HarabelForProtected extends Shpend {
6     public static void main(String[] args) {
7         HarabelForProtected harabel = new HarabelForProtected();
8         harabel.cicerima = "ciu ciu ciu";
9         harabel.shfaq();
10    }
11 }
```

Fig. 9 Klasa *HarabelForProtected* (Protected Access Modifier)

Shohim si do të dukej nëse nuk do të trashëgonim klasën *Shpend*. Variabli *cicerima* dhe *shfaq()* nuk do të mund të përdroreshin nga objekti *harabel*. Krijojmë kështu objektin e tipit *Shpend* pasi klasa *Shpend* ka access modifier *public*. Nga objekti **shpend** mundohemi t'i japim vlerë variablit *cicerima* apo të thërrasim metodën *shfaq()*, por na shfaqet mesazhi **"'cicerima' has protected access in 'ProtectedAccessModifier.Shpend'"** dhe vet IDE-ja na sugjeron të ndryshojmë access modifier në *public*.



```
1 package OtherPackageForAccessModifiers;
2
3 import ProtectedAccessModifier.Shpend;
4
5 public class HarabelForProtected {
6     public static void main(String[] args) {
7
8         Shpend shpend = new Shpend();
9         shpend.cicerima = "ciu ciu ciu";
10        shpend.shfaq();
11    }
12 }
13
```

'cicerima' has protected access in 'ProtectedAccessModifier.Shpend'

Create field 'cicerima' in 'Shpend' Alt+Shift+Enter More actions... Alt+Enter

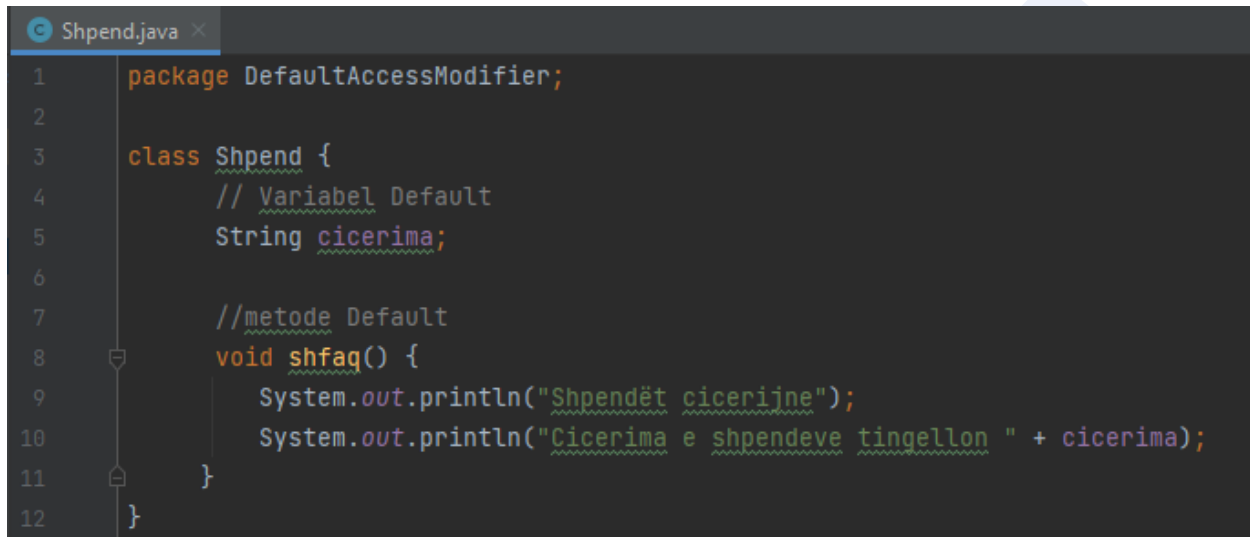
ProtectedAccessModifier.Shpend
protected String cicerima

Fig. 10 Klasa *HarabelForProtected* pa trashëgimi (Protected Access Modifier)

3. Default Access Modifier

Nëse nuk specifikojmë në mënyrë eksplicite një access modifier për klasat, metodat, variablat etj, atëherë si default konsiderohet *Default Access Modifier*. **Default** lejon aksesimin vetëm brenda të njëjtës package (nga subclasses dhe jo-subclasses). Shohim shembullin e mëposhtëm.

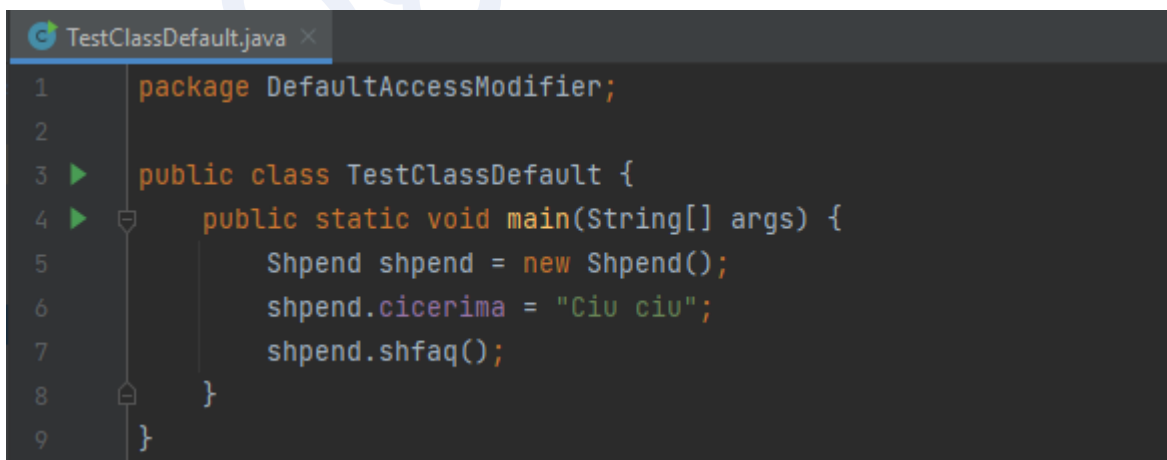
Në *package DefaultAccessModifier* kemi klasën **Shpend** në të cilën nuk kemi specifikuar një Access Modifier kështu që by default access modifier i kësaj klase është **Default**. Gjithashtu, njëjloj dhe për variablin e instancës cicerima dhe metodën shfaq().



```
1 package DefaultAccessModifier;
2
3 class Shpend {
4     // Variabel Default
5     String cicerima;
6
7     //metode Default
8     void shfaq() {
9         System.out.println("Shpendët cicerijne");
10        System.out.println("Cicerima e shpendeve tingellon " + cicerima);
11    }
12 }
```

Fig. 11 Klasa Shpend (Default Access Modifier)

Në klasën **TestClassDefault**, e cila ndodhet në të njëjtën *package* si klasa Shpend me *access modifier Default*, krijojmë objektin e tipit Shpend dhe replikojmë kodin si më sipër. Nëse i bëjmë run marrim output-in si më parë. Pra, brenda të njëjtës *package* nuk ka ndryshuar asgjë. Mund ta aksesojmë si normalisht.



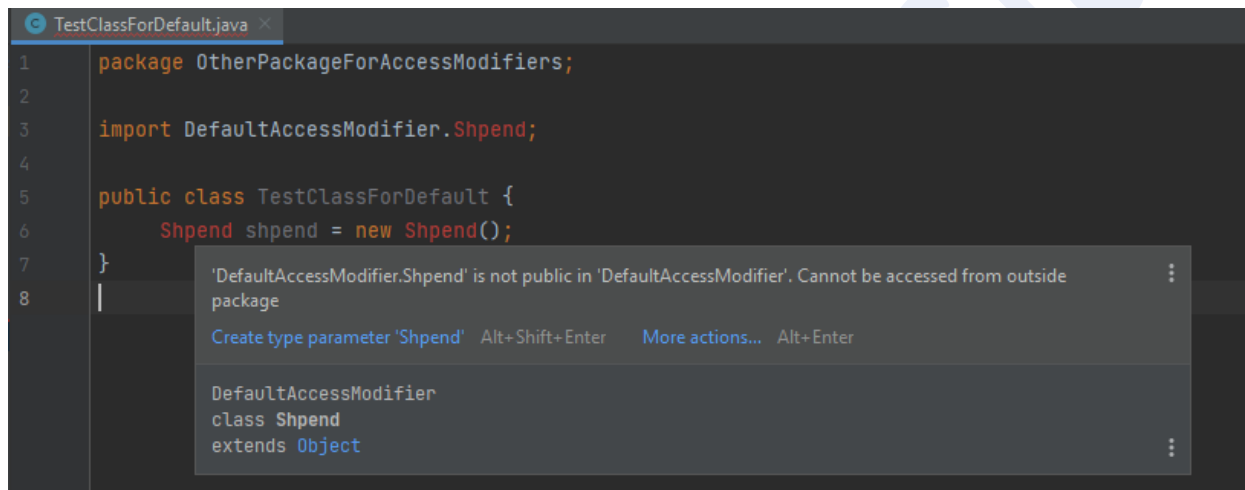
```
1 package DefaultAccessModifier;
2
3 public class TestClassDefault {
4     public static void main(String[] args) {
5         Shpend shpend = new Shpend();
6         shpend.cicerima = "Ciu ciu";
7         shpend.shfaq();
8     }
9 }
```

Fig. 12 Klasa TestClassDefault (Default Access Modifier)

```
Shpendët cicerijne  
Cicerima e shpendeve tingellon Ciu ciu
```

Fig. 13 Output TestClassDefault (Default Access Modifier)

Në një *package* tjetër (në këtë rast **OtherPackageForAccessModifiers**) krijojmë një klasë **TestClassForDefault** si më poshtë. Mundohemi të krijojmë një objekt të tipit Shpend duke importuar klasën Shpend nga package **DefaultAccessModifier**, por na shfaqet mesazhi *"DefaultAccessModifier.Shpend" is not public in "DefaultAccessModifier". Cannot be accessed from outside package"*. Pra, nuk mund të aksesohet. Po sikur të trashëgojmë klasën Shpend? Përsëri do të na shfaqet i njëjti mesazh si më parë, si e shihni dhe në fotot më poshtë.



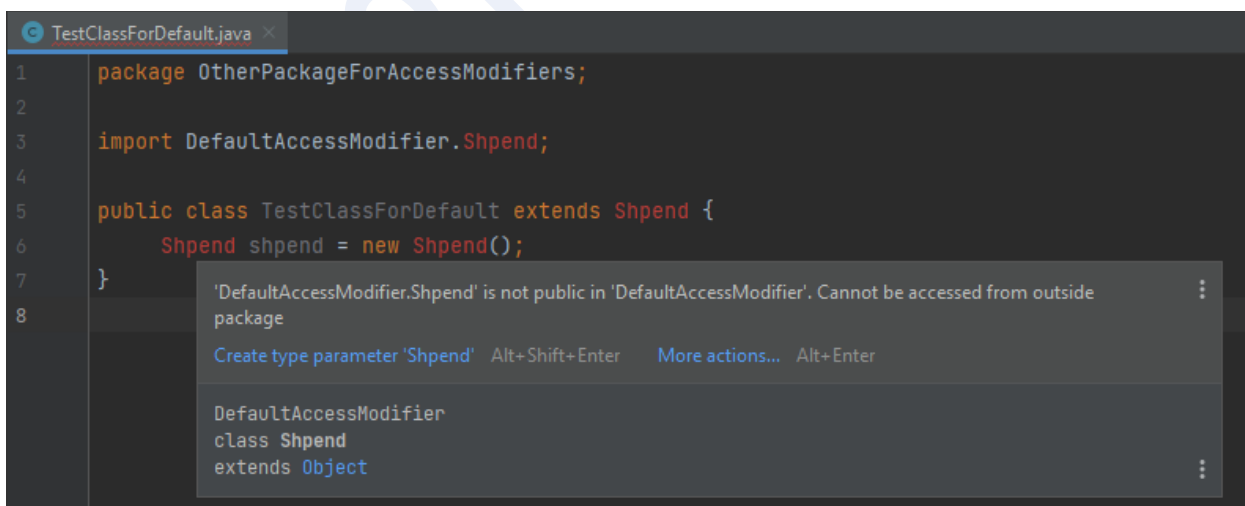
```
1 package OtherPackageForAccessModifiers;  
2  
3 import DefaultAccessModifier.Shpend;  
4  
5 public class TestClassForDefault {  
6     Shpend shpend = new Shpend();  
7 }  
8
```

'DefaultAccessModifier.Shpend' is not public in 'DefaultAccessModifier'. Cannot be accessed from outside package

Create type parameter 'Shpend' Alt+Shift+Enter More actions... Alt+Enter

DefaultAccessModifier
class Shpend
extends Object

Fig. 14 Klasa TestClassForDefault (Default Access Modifier)



```
1 package OtherPackageForAccessModifiers;  
2  
3 import DefaultAccessModifier.Shpend;  
4  
5 public class TestClassForDefault extends Shpend {  
6     Shpend shpend = new Shpend();  
7 }  
8
```

'DefaultAccessModifier.Shpend' is not public in 'DefaultAccessModifier'. Cannot be accessed from outside package

Create type parameter 'Shpend' Alt+Shift+Enter More actions... Alt+Enter

DefaultAccessModifier
class Shpend
extends Object

Fig. 15 Klasa TestClassForDefault me trashëgimi (Default Access Modifier)

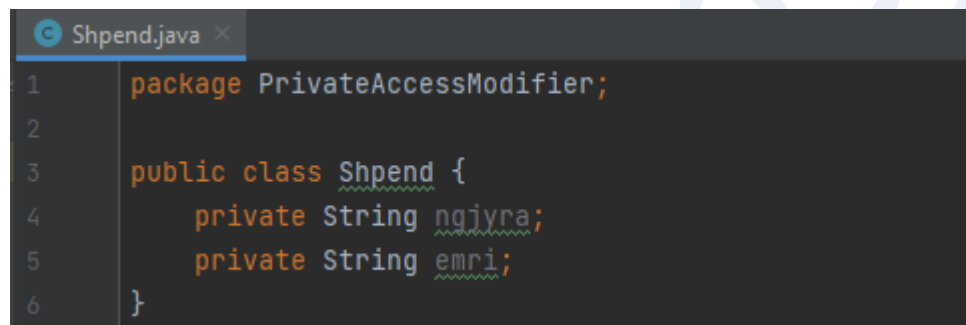
4. Private Access Modifier

Kur metodat apo variablat deklarohen private ato mund të aksesohen vetëm brenda klasës ku janë deklaruar.



Një klasë apo interface nuk mund të deklarohet private (logjikojmë si më sipër me access modifier protected). Nëse deklarojmë një klasë private atëherë nuk mund ta përdorim dot.

Shohim shembullin në vijim. Në *package* **PrivateAccessModifier** krijojmë klasën **Shpend** me variablat e instancës *private* (ngjyra, emri). Pa shtuar akoma asgjë tjetër në klasën tjetër brenda këtij *package* **TestClassPrivate** krijojmë objektin e tipit Shpend. Mundohemi t'i japim vlera dy variablaive të instancës brenda klasës **Shpend** (ngjyra & emri), por shohim se na shfaqen mesazhet "'ngjyra' has private access in 'PrivateAccessModifier.Shpend'" & "'emri' has private access in 'PrivateAccessModifier.Shpend'".



```
1 package PrivateAccessModifier;
2
3 public class Shpend {
4     private String ngjyra;
5     private String emri;
6 }
```

Fig. 16 Klasa Shpend pa Getters & Setters (Private Access Modifier)



```
1 package PrivateAccessModifier;
2
3 public class TestClassPrivate {
4     public static void main(String[] args) {
5         Shpend shpend = new Shpend();
6         shpend.ngjyra = "e zeze";
7         shpend.emri = "nje emer";
8     }
9 }
10
```

'emri' has private access in 'PrivateAccessModifier.Shpend'

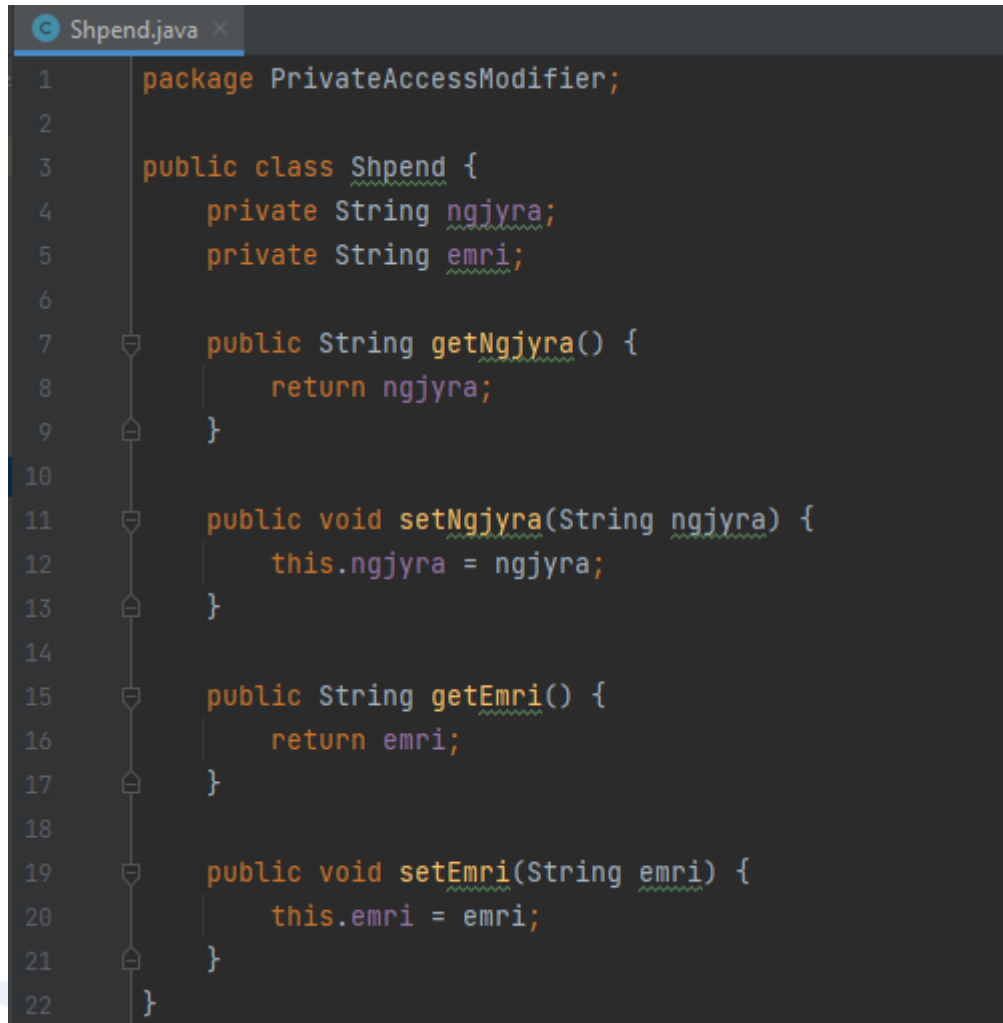
Create field 'emri' in 'Shpend' Alt+Shift+Enter More actions... Alt+Enter

PrivateAccessModifier.Shpend
private String emri

Fig. 17 Klasa TestClassPrivate (Private Access Modifier)

Por, pse përdorim *access modifier private* nëse nuk mund t'i aksesojmë dot fushat jashtë klasës ku deklarohen? Ju kujtohet koncepti i Encapsulation në temën e OOP? Nëse doni ta rikujtoni klikoni [këtu](#). Në përgjithësi, *access modifier private* e përdorim kur nuk duam të lejojmë klasat/përdoruesit

e jashtëm të hyjnë dhe të modifikojnë të dhënat tona. Si mund t'i aksesojmë këto fusha jashtë klasës Shpend? Duke përdorur **Getters** dhe **Setters**. Këto janë metoda publike ku *Setters* përdoren për t'i vendosur vlerë variablit, kurse *Getters* për të marrë vlerën e një variabli të caktuar. Bëjmë këto ndryshime në klasën **Shpend** dhe më pas në klasën **TestClassPrivate** i vendosim vlerat dhe i printojmë ato. Shohim në figurat më poshtë.



```
1 package PrivateAccessModifier;
2
3 public class Shpend {
4     private String ngjyra;
5     private String emri;
6
7     public String getNgjyra() {
8         return ngjyra;
9     }
10
11     public void setNgjyra(String ngjyra) {
12         this.ngjyra = ngjyra;
13     }
14
15     public String getEmri() {
16         return emri;
17     }
18
19     public void setEmri(String emri) {
20         this.emri = emri;
21     }
22 }
```

Fig. 18 Klasa Shpend me Getters & Setters (Private Access Modifier)

```

1 package PrivateAccessModifier;
2
3 public class TestClassPrivate {
4     public static void main(String[] args) {
5         Shpend shpend = new Shpend();
6
7         // Vendosim ngjyren dhe emrin
8         shpend.setNgjyra("e zeze");
9         shpend.setEmri("nje emer");
10
11        // Printojme vlerat e sapo vendosura
12        System.out.println("Ngjyra e vendosur eshte: " + shpend.getNgjyra());
13        System.out.println("Emri i vendosur eshte: " + shpend.getEmri());
14    }
15 }

```

Fig. 19 Klasa TestClassPrivate duke perdorur Getters & Setters (Private Access Modifier)

Pasi bëjmë run klasën TestClassPrivate tani do të marrim output-in me vlerat e ngjyrës dhe emrit si më poshtë.

```

Ngjyra e vendosur eshte: e zeze
Emri i vendosur eshte: nje emer

```

Fig. 20 Output TestClassPrivate duke perdorur Getters & Setters (Private Access Modifier)

Më tej mund të provoni të krijoni një klasë test në *package* **OtherPackageForAccessModifiers** dhe të testoni krijimin e objektit të tipit Shpend, si në shembujt më sipër.

Përfundim

Në këtë tabelë kemi një përmbledhje për secilin *Access Modifier*.

Access Modifier	Brenda klasës	Brenda <i>package</i>	Jashtë <i>package</i> vetëm nga <i>subclass</i>	Jashtë <i>package</i>
Public	Po	Po	Po	Po
Protected	Po	Po	Po	Jo
Default	Po	Po	Jo	Jo
Private	Po	Jo	Jo	Jo

Në këtë artikull mësuam se çfarë janë *access modifiers* në Java, cilët janë dhe përse përdoren. Është praktikë e mirë të përdoret niveli më kufizues i Access Modifiers për çdo variabël, metodë

etj. për të parandaluar keqpërdorimin e tyre. Gjithmonë duhet të përdorim *private access modifier* nëse nuk ka një arsye të mirë për të mos e përdorur. *Access modifier public* duhet të përdoret vetëm nëse një pjesëtar është pjesë e një API. Si gjithmonë, shembujt e kodit janë në dispozicion në Github (folder **AccessModifiersExamples**).

Nëse keni pyetje të mëtejshme, mos ngurroni të pyesni në instagram [@programuesja](#) ose në adresën email programuesja@gmail.com.

Programuesja