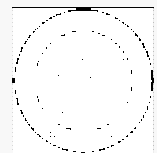


# Il linguaggio Java

*Il controllo degli accessi alle classi ed ai membri di una classe*



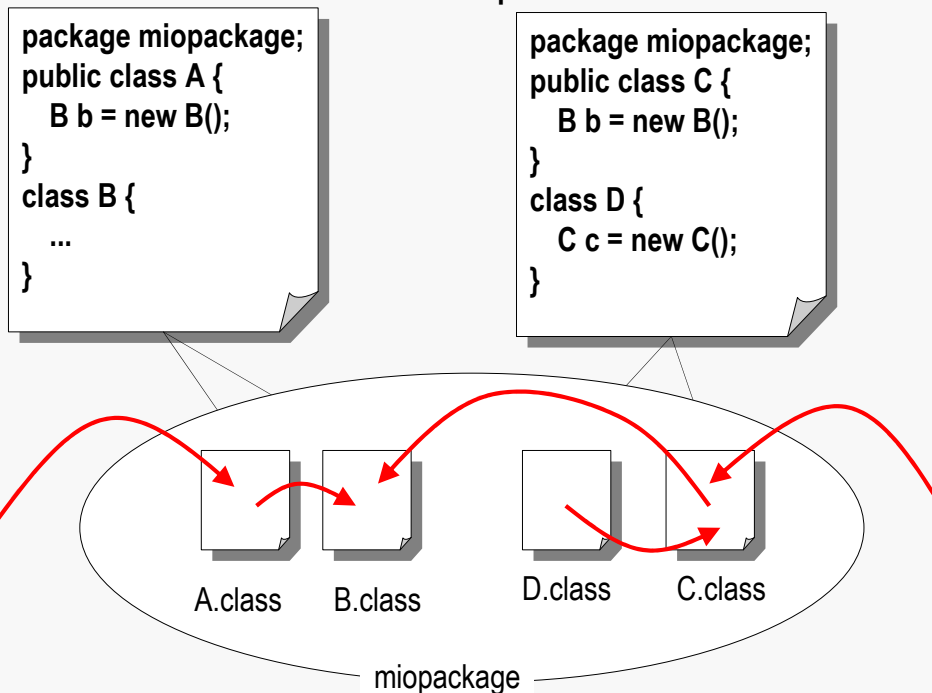
# Controllo degli accessi

*Accesso alle classi*

# Accesso alle classi



unità di compilazione



TIGA

3

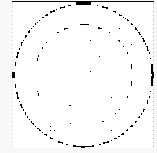
# Accesso alle classi



- *Accesso pubblico.*
  - Il modificatore di accesso **public** specifica che la classe è accessibile da qualunque parte del programma
- *Accesso a livello di package.*
  - L'assenza del modificatore di accesso **public** specifica che la classe è accessibile solo all'interno del package
  - In questo caso è buona norma
    - definire **private** i campi membro
    - dare alle funzioni membro l'accessibilità a livello di package
- Non è possibile dichiarare **private** o **protected** una classe

TIGA

4



# Controllo degli accessi

## *Accesso ai membri di una classe*

### ***Motivazioni***

---



- *Incapsulamento*

L'utilizzatore di una classe non deve

- vedere come la classe è stata implementata
- interferire con l'implementazione della classe

- *Sicurezza*

Alcuni dati servono per l'implementazione della classe ma non devono essere accessibili all'utilizzatore (ad oggetti di altre classi)

# Livelli e specificatori di accesso



- *Quattro livelli di accesso membri di una classe*
  - privato
  - pubblico
  - protetto
  - pacchetto
- L'accesso ai membri di una classe è controllato attraverso gli *specificatori di accesso*
  - **private**
  - **public**
  - **protected**
  - *nessun specificatore*

# La tabella di accesso



	LIVELLO DI ACCESSO			
<i>specificatore</i>	<i>classe</i>	<i>sottoclasse</i>	<i>package</i>	<i>pubblico</i>
<b>private</b>	X			
<b>protected</b>	X	X*	X	
<b>public</b>	X	X	X	X
<b>nessuno</b>	X		X	

# Livello di accesso pubblico e privato



- Livello di accesso pubblico (parola chiave **public**)

Un membro pubblico non è un segreto per nessuno

- Un membro pubblico di una classe è accessibile da qualunque classe in qualunque pacchetto

- Livello di accesso privato (parola chiave **private**)

Un membro privato è un segreto personale

- Un membro privato è accessibile solo dalla classe nel quale è definito (restrizione a livello di classe)

*Due oggetti istanza della stessa classe, hanno accesso l'uno alle variabili private dell'altro*

# Accesso privato: accesso da altri tipi



```
class A {
    private int varPriv;
    private void metodoPrivato() {
        System.out.println("A: Metodo privato");
    }
}
```

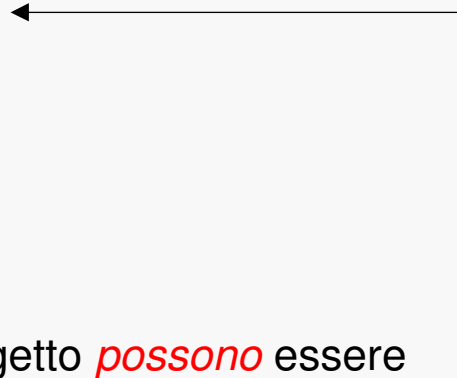
```
class B {
    void metodoAccesso() {
        A a = new A();
        a.varPriv = 1;           // illegale
        a.metodoPrivato();      // illegale
    }
}
```

I membri privati di un oggetto **non possono** essere acceduti da oggetti di tipo diverso

## Accesso privato: accesso dallo stesso tipo



```
class A {
    private int varPriv;
    boolean isEqualTo(A altro) {
        if (this.varPriv == altro.varPriv)
            return true;
        else
            return false;
    }
}
```



I membri privati di un oggetto *possono* essere acceduti da oggetti dello stesso tipo

## Livello di accesso protetto



- Il livello di accesso protetto (parola chiave **protected**)

Un membro protetto è un segreto di famiglia noto anche ad alcuni amici fidati

- Un membro protetto è accessibile dalla classe in cui è definito, dalle sue sottoclassi<sup>(\*)</sup> e da tutte le classi che appartengono allo stesso package

<sup>(\*)</sup> Con una importante limitazione. Vedere più avanti.

## Livello protetto: accesso dal pacchetto



Un membro protetto definito in una classe è accessibile da tutte le classi che appartengono allo stesso package della classe in cui il membro è definito

```
package Bianco;

public class A {
    protected int pv;
    protected void pm() {
        System.out.println("metodo protetto");
    }
}
```

```
package Bianco;

public class B {
    void accesso() {
        A a = new A();
        a.pv = 1; // legale
        a.pm(); // legale
    }
}
```

## Livello protetto: accesso da sottoclasse



Un membro protetto di una classe è accessibile da tutte le sottoclassi (anche appartenenti a diversi pacchetti) ma **una sottoclasse può accedere solo al membro protetto dei suoi oggetti o di quelli delle sue sottoclassi**

```
package Bianco;

public class A {
    protected int pva;
    protected void pm() {
        System.out.println("metodo
                               protetto");
    }
}
```

```
package Nero;
import Bianco;

public class C extend A {
    void accesso(A a, C c) {
        a.pva = 1; // illegale
        a.pm(); // illegale
        c.pva = 1; // legale
        c.pm(); // legale
    }
}
```

## Livello protetto: accesso da sottoclasse



```
package Bianco;
import Nero;

public class A {
    protected int pva;
    protected void pm() {
        C c = new C();
        c.pva = 10; // legale
        c.pvc = 1; // illegale
    }
}
```

```
package Nero;
import Bianco;

public class C extend A {
    protected int pvc;
    void accesso(A a, C c) {
        a.pva = 1; // illegale
        a.pm(); // illegale
        c.pva = 1; // legale
        c.pm(); // legale
    }
}
```

## Livello pacchetto



Un membro di una classe senza specificatori di accesso è accessibile da tutte le classi che appartengono allo stesso pacchetto della classe

Un membro senza specificatori è un segreto noto ad un ristretto gruppo di amici fidati ma non alla propria famiglia

```
package Bianco;

public class A {
    int packv;
    void packm() {
        System.out.println("metodo protetto");
    }
}
```

```
package Bianco;

public class B {
    void accesso() {
        A a = new A();
        a.packv = 1; // ok
        a.packm(); // ok
    }
}
```