Classi interne

La possibilità di creare delle classi innestate amplia il discorso sulle regole di visibilità.

Le classi interne non statiche hanno le seguenti proprietà:

1) Privilegi di visibilità tra classe contenitrice e altre innestate. Forte comunicazione

2) Restrizioni di visibilità tra le classi esterne e quella contenitrice. Incapsulamento

3) Riferimento implicito ad un oggetto della classe contenitrice. Ogni classe interna conosce l'oggetto creato dalla contenitrice

Restrizioni di visibilità

La classe più esterna è la top-level, e le classi interne possono avere tutte le quattro visibilità.

La visibilità della classe interna X dalla quali classi possono usarla

Consideriamo il seguente esempio:

```
public class A {
    private class B {
        ...
    }
    class C {
        ...
    }
}
```

Fuori A, i nomi completi di B e C sono A.B e A.C

La classe B **non è visibile al di fuori di A** C può vedere B

La classe C è visibile a tutte le classi che si trovano nello stesso pacchetto di A

Tra classi interne non vige nessuna restrizione di visibilità

Relazione tra oggetti

Ogni oggetto di inner class non statica ha un riferimento implicito ad un oggetto della classe contenitrice, inizializzato alla creazione dell'oggetto

Se B è interna ad A → dentro B posso indicare il rif. implicito A.this

↓

Facoltativo

B può accedere a campi e metodi anche senza A.this

Esempio:

```
public class A {
    private int x;

    public class B {
        private int y;
        public void stampami() {
            System.out.println(A.this.x);   // uso del riferimento implicito
            System.out.println(y);
        }
    }
}
```

Contesto statico

Si trova dove è presente il modificatore static

- Sintesi un metodo static;

- Dentro un blocco static;
- Definizione di un attributo static

Il resto è contesto non statico

Consideriamo B interna ad A:

- In un contesto non statico di A, se creo un oggetto di tipo B, il riferimento implicito verrà inizializzato con il valore attuale di this;
- Negli esempi uso new

<rifer. ad oggetto di tipo A> . new B(...)

Esempio: creazione di un oggetto di classe interna in un *contesto non statico* della classe esterna

```
public class A {
    private int x;

    public class B {
        private int y;
        public void stampami() {
            System.out.println(A.this.x);    // uso del riferimento implicito
            System.out.println(y);
        }
    }

    public B makeB(int val) {
        B b = new B();     // il nuovo oggetto B è legato a this
        b.y = val;         // privilegi di visibilità
        return b;
    }
}
```

Esempio: creazione di un oggetto di classe interna in un contesto *statico* della classe esterna

```
public class A {
    private int x;

    public class B {
        private int y;
        public void stampami() {
            System.out.println(A.this.x);    // uso del riferimento implicito
            System.out.println(y);
        }
    }

    public static void main(String[] args) {
        B b = new B();     // errore di compilazione
        A a = new A();
        B b = a.new B();   // OK
    }
}
```

Classi interne statiche

Le classi interne statiche non possiedono il riferimento implicito

Ricapitolazione delle proprietà di visibilità delle inner class non statiche

1) Privilegi di visibilità tra classe contenitrice e altre innestate. Forte comunicazione

2) Restrizioni di visibilità tra le classi esterne e quella contenitrice. Incapsulamento

3) Riferimento implicito ad un oggetto della classe contenitrice. Ogni classe interna conosce l'oggetto creato dalla contenitrice

Le classi interne statiche godono delle prop 1 e 2