

```
        return this.eta;
    }
    public int setEta(int eta){
        this.eta = eta;
    }
}
```

Ogni costante deve essere definita con i modificatori “static final” per garantire che il valore non sia modificato e sia accessibile staticamente.

Esempio:

```
static final int key = 1;
```

7.2.10.2 Visibilità

Classi, metodi e variabili devono essere esplicitamente marcate come private, protette o pubbliche, per limitare il livello di accesso da parte di altri oggetti. Laddove non è necessario esporre delle funzionalità, deve essere impostato un livello di visibilità più ristretto, al fine di evitare l'esposizione di strutture interne.

7.2.10.3 Modificatori

Ove possibile, è necessario rendere le classi, i metodi e le variabili di tipo “final”.

L'utilizzo di questo modificatore consente di aumentare l'efficienza del programma in fase di esecuzione, in quanto non consente il "late binding".

Esempio:

```
public final class MyFinalClass {
    [...]
}

public class MyClass {
    final int myConst = 123;
    [...]
}

public class MyClass {
    [...]
    public final void stopOverriding() {
        [...]
    }
}
```

Le variabili di tipo static dovrebbero essere limitate allo stretto necessario, poiché la loro visibilità prescinde dal ciclo di vita degli oggetti e sono perciò meno controllabili. Le variabili statiche sono globali e in un ambiente multitutente possono essere modificate da più thread, con risultati imprevedibili.

7.2.10.4 Utilizzo degli oggetti mutevoli

Un metodo non dovrebbe mai tornare oggetti mutevoli (ad esempio array, liste, vettori, date, etc..) e non dovrebbero mai essere memorizzati internamente in modo diretto (dovrebbero invece essere opportunamente clonati). Si tratta di oggetti raggiunti per indirizzo, per cui tornare un array privato da un metodo pubblico, esporrebbe i dati a possibili manipolazioni da parte di un attaccante.

Esempio 1:

Forma non corretta:

```
public Date getDate() {
    return fDate;
}
```

Forma corretta:

```
public Date getDate() {
    return new Date(fDate.getTime());
}
```