

JAVA 8 BEST PRACTICES CHEAT SHEET

Lo Sviluppatore

Default Methods Fai evolvere le Interfacce

```
@FunctionalInterface
public interface IA{
    public void f();

    // Un metodo di default
    default public void a(){
        System.out.println("Ciao da a");
    }
}

public class B implements IA{
    public void f(){
        int a = 5;
    }
    ...
    {
        /* invocazione del metodo di
        default. Stamperà sullo schermo "Ciao
        da a" */
        new B().a()
    }
}
```

Lambdas Sintassi: (parametri)->espressione (parametri)->{istruzioni;}

```
// prende in input due interi e
restituisce la somma
(int x, int y) -> x + y

// prende in input una stringa e non
restituisce nulla
(String s) -> {
    System.out.println("Benvenuto "+s);
}

// map: converte ogni elemento in
qualcosa'altro
map(e -> e.hashCode())

// filter: fa passare solo alcuni
elementi
filter(e ->((e.hashCode() % 2) == 0))
```

java.util.Optional Un contenitore per possibili valori null

```
// crea un Optional
Optional<Integer> op =
Optional.ofNullable(valore);

// elabora l'optional
op.map(s->"Valore="+s);

// esegue il blocco se è presente un
valore non nullo
op.ifPresent(s->System.out::println)

// restituisce il valore se c'è o
lancia un'eccezione
op.get();

// restituisce il valore se c'è o la
stringa "Ciao" altrimenti
op.orElse("Ciao");
```

Regole Empiriche

- "Un metodo di default per Interfaccia
- "Non estendere le Interfacce Funzionali
- "Solo implementazioni conservative (compatibilità con le Interfacce preesistenti)

- " Espressioni e Statement
- " Fai il refactor usando i method reference
- " Concatena le Lambda anziché fare blocchi di istruzioni troppo lunghi

- " Campi : usare plain objects
- " Parametri dei metodi: usare plain objects
- " Valori di ritorno dei metodi : usare gli Optional
- " Usare `orElse()` invece di `get()`