

Dynamic dispatch con interfacce in Java

Il sistema utilizzato per il dispatching efficiente di metodi nel caso di ereditarietà singola non funziona con le interfacce: non si può garantire che tutte le classi che implementano un'interfaccia abbiano i metodi nella stessa posizione.

```
interface I { foo(); }    // 0
interface J { bar(); }    // 0
class C { foo(); bar(); } // non entrambi 0!
```

Soluzione:

- ogni classe ha una *itable* che contiene puntatori ai metodi di tutte le interfacce implementate, accessibile tramite un puntatore nella tabella dei metodi;
- quando viene chiamato un metodo appartenente a un'interfaccia è necessario eseguire una ricerca nella itable (con memoizzazione per rendere più veloci le chiamate successive);
- le invocazioni di metodi di classe continuano ad usare `invokevirtual`.

Da Java 8 le interfacce possono contenere implementazioni di default, di fatto portando tutte le problematiche dell'ereditarietà multipla. Invece di risolverle a livello di linguaggio, il compilatore dà errore quando si potrebbe verificare ambiguità.