Programmazione in Java e gestione della grafica

Lezione 16

TEST

- TestMar: 25 marzo 2013

Polimorfismo

- Consente di “programmare il caso generale”
- La stessa chiamata può produrre “risultati diversi”

Esempi di polimorfismo

- Quando un programma chiama un metodo attraverso una variabile superclasse, viene effettivamente chiamata la corretta versione del metodo nella sottoclasse corrispondente al tipo di variabile referenziata.
- E' facile aggiungere nuove classi con minime modifiche al codice.

Polimorfismo

- Posso assegnare ad un riferimento della superclasse un oggetto della sottoclasse
  - Questo è possibile perchè un oggetto della sottoclasse *è un* oggetto della superclasse.
  - Quando chiamo un metodo da quel riferimento, è il tipo dell'oggetto effettivamente referenziato che determina quale metodo è chiamato.
- Posso assegnare ad un riferimento della sottoclasseun oggetto della superclasse solo se dopo aver eseguito **downcast**

Esempio direttamente al compilatore

dal libro di testo cap. 10

Classi e metodi astratti

- Classi che sono troppo generali per c reare oggetti reali
- Usate come superclassi astratte per dichiarare delle variabili di riferimento
- Molte gerarchie di ereditarietà hanno superclassi astratte ai livelli più alti
- Parola chiave `abstract`
  - Uso per dichiarare una classe `abstract`
  - Uso per dichiarare un metodo `abstract`
    - le classi astratte contengono uno o più metodi astratti
    - Le classi concrete sottostanti devono riscrivere TUTTI I metodi astratti della superclasse classe astratta

Operatore `instanceof`

- `instanceof`
  - Determina se un oggetto è un istanza di un certo tipo (classe)

Operatore di Downcasting

- Downcasting
  - Converte il riferimento ad una superclasse ad un riferimento alla sottoclasse Convert a reference to a superclass to a reference to a subclass
  - (permesso solo si tratta di una sottoclasse)
- `getClass` metodo di `Object`
  - Restituisce un oggetto di tipo `Class`
- `getName` metodo of `Class`
  - Restituisce il nome della classe

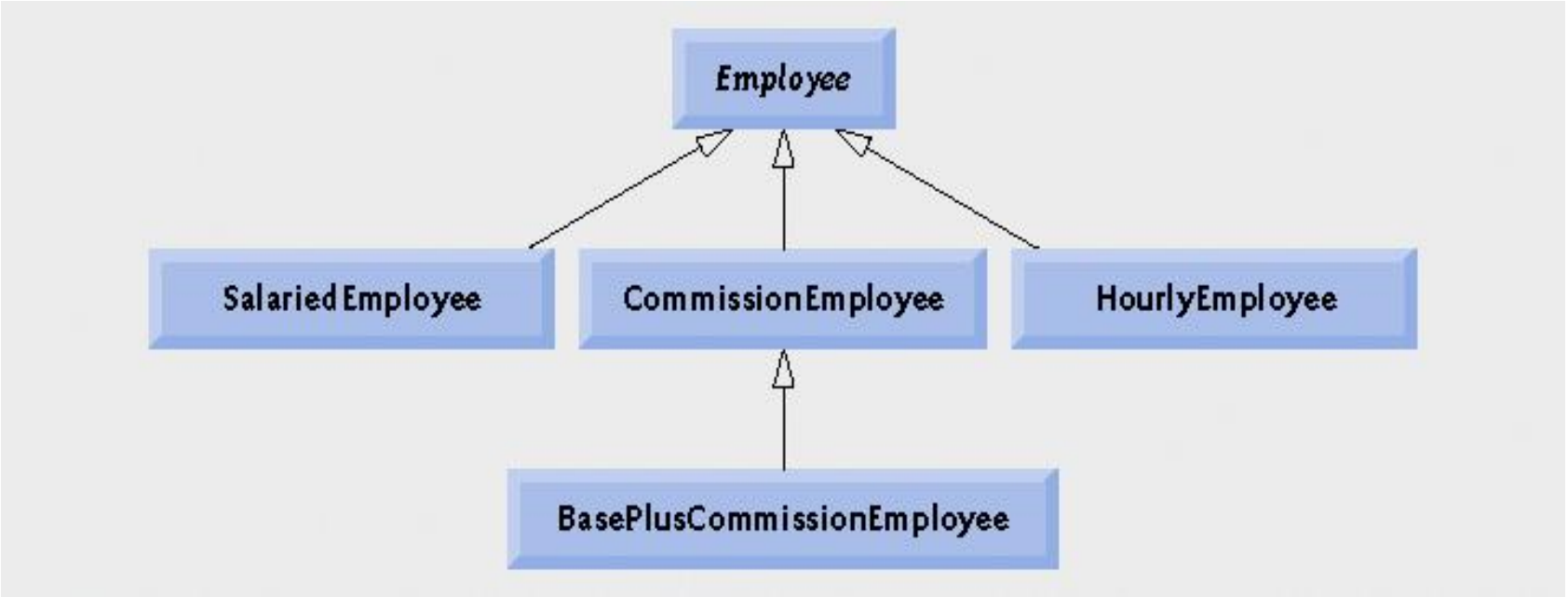

Diagramma della gerarchia per la classe `Employee`

Esempio direttamente al compilatore

dal libro di testo cap. 10

Riassumendo….

- Regole di assegnamento tra variabili di superclasse e di sottoclasse
  - Assegnare un riferimento alla superclasse ad una variabile della superclasse OK
  - Assegnare un riferimento alla sottoclasse ad una variabile della sottoclasse OK
  - Assegnare un riferimento alla sottoclasse ad una variabile della superclasse OK (perchè contenuta)
    - Riferirsi ai membri specifici della sottoclasse attraverso variabili superclasse dà errore di compilazione
  - Assegnare un riferimento alla superclasse ad una variabile della sottoclasse dà errore di compilazione
    - Bisogna ricorrere al Downcasting

10.6 Metodi e Classi `final`

- Metodi `final`
  - Non si può fare overridden in una sottoclasse
  - I metodi `private` e `static` sono implicitamente `final`
- Classi `final`
  - Non possono essere estese da una sottoclasse

    ERRORE COMPILAZIONE!
  - Tutti I metodi di una classe `final` sono implicitamente `final`

Osservazione

- La maggior parte di classi delle API Java non sono dichirate `final`. Questo permette di sfruttarle pienamente con l'ereditarietà e il polimorfismo .

- In alcuni casi è importante dichiarare certe classi `final` (tipicamente per motivi di sicurezza).
- La classe `string` è dichiarata `final`