Insegnamento di Laboratorio di algoritmi e strutture dati

L'interfaccia in Java

Roberto Posenato

ver. 0.7, 29/01/2008

L'interfaccia in Java

Premessa

- Si assume che il concetto generale di interfaccia sia noto agli studenti in quanto già presentato al corso di Architetture Software o Ingegneria del software.
- Si assume anche che il concetto di classe astratta in generale e in Java sia noto in quanto già presentato al corso di Programmazione.
- In questa lezione si precisa cosa significa dichiarare e implementare un'interfaccia nel linguaggio Java.

L'interfaccia in Java

Esigenza...

- In diversi contesti è necessario suddividere un progetto software in sottoprogetti assegnandoli a gruppi diversi.
- La cooperazione di questi gruppi deve avvenire mediante un accordo di come le diverse componenti interagiscono fra loro.
- Ciascun gruppo deve poter sviluppare le proprie componenti senza conoscere come sono implementate quelle degli altri gruppi.
- **Informalmente, le interfacce rappresentano questi accordi.**
- L'insieme delle interfacce di una componente software è anche detto application programming interface (API).

L'interfaccia in Java

Definizione

Definizione 1 (Interfaccia)

Un'interfaccia, (keyword `interface`), definisce un *tipo* in modo simile a `class` e può contenere solamente:

- *costanti*
  Esempio: `public final int MAX_SIZE = 255;`
- *segnature/firme di metodi*
  Esempio:
  `public void setSize(int);` *//Il corpo {} non si mette!*
- *tipi innestati*
  Esempio: `public class SottoClasse {...}`

L'interfaccia in Java

Esempio

Esempio di interfaccia

```
public interface Lista { //Selettore: public o nulla!
    int MAX_SIZE=100; //Selettore implicito: public static final
    boolean eVuota(); //Selettore implicito: public
    int lunghezza();
    Object elemento(int k);
    void cancella(int k);
    void inserisci(int k, Object o);
}
```

Nota!

- Le interfacce non possono essere istanziate

  ```
  Lista miaLista = new Lista(); //Errore!
  ```

- Le interfacce possono essere solo implementate da classi o estese da altre interfacce.

L'interfaccia in Java

Implementazione

Una classe implementa un'interfaccia se:

1. Dichiara di implementarla

```
public class MiaLista implements Lista { ... };
```

2. Ridichiara ogni metodo dell'interfaccia (identica segnatura) aggiungendo il corpo

```
public class MiaLista implements Lista {
    private Object[] lista;
    ...
    public boolean eVuota() { //Notare public!
        return lista!=null && lista.length>0;
    }
    ...
    public void nonInLista() {}; //Non è in Lista!
}
```

3. Una classe che implementa un'interfaccia si dice essere in relazione di realizzazione con l'interfaccia.

L'interfaccia in Java

Uso

Nota!

- Una classe che implementa un'interfaccia può essere usata per definire oggetti del tipo definito dall'interfaccia:

```
/*MiaLista implementa Lista*/
Lista miaLista = new MiaLista();
```

- L'oggetto miaLista può usare tutti i metodi di Lista, ma non di MiaLista:

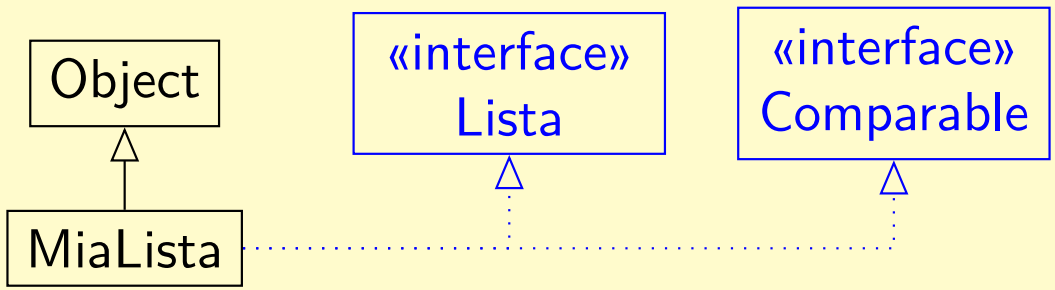
```
miaLista.eVuota();  //OK
miaLista.nonInLista();  //NO: non è presente in Lista
```

L'interfaccia in Java

Relazione di ereditarietà e relazione di realizzazione

Nota!

- Si osservi che il concetto di interfaccia è cosa diversa del concetto di ereditariertà.
- La relazione di ereditarietà stabilisce una gerarchia tra le classi rappresentata da un albero con radice.
- La relazione di realizzazione stabilisce una gerarchia che può NON essere un albero!



L'interfaccia in Java

Specifiche

Nota!

- Ok, un'interfaccia è una definizione sintattica di un tipo... ma come faccio implementare correttamente un'interfaccia se non conosco cosa devono fare i suoi metodi?
- NON c'è modo per garantire che un'implementazione corrisponda correttamente a ciò che si vuole con un'interfaccia.
- Si può corredare l'interfaccia di Javadoc che descrive in modo informale il significato dei metodi.
  La documentazione di ciascun metodo è detta specifica o contratto.

L'interfaccia in Java

Esempio

Esempio di interfaccia con specifica

```
/**
 * Descrive i metodi fondamentali per il tipo Lista.
 * @author Roberto Posenato
 */
public interface Lista {
    /**
     * Massimo numero di elementi ammessi in una lista.
     */
    int MAX_SIZE = 100;

    /**
     * Indica se la lista è vuota.
     * @return true se è vuota, false altrimenti.
     */
    boolean eVuota();
```

L'interfaccia in Java

Esempio

continua

```
    /**
     * Restituisce il numero degli oggetti presenti.
     * @return intero maggiore uguale di 0.
     */
    int lunghezza();

    /**
     * Restituisce l'oggetto di indice k se esiste,
     * null altrimenti.
     * @param k indice elemento ricercato.
     * @return oggetto di indice k se (0≤k<lunghezza()),
     *         null altrimenti.
     */
    Object elemento(int k);
    ...
}
```

L'interfaccia in Java

Osservazioni finali

- Un'interfaccia in Java specifica un Tipo di Dati Astratto (ADT);
- Data un'interfaccia si possono avere più classi che la implementano (come vedremo dopo);
- Un metodo che usa le interfacce come tipi per i suoi parametri può essere usato su oggetti di diversi classi! (Quali?)

L'interfaccia in Java

Tocca a voi!

Esercizio 1

1. Studiare le differenze che ci sono tra le Interfacce e le Abstract class.
   Vedi `http://java.sun.com/docs/books/tutorial/java/IandI/abstract.html`
2. Cercare la specifica dell'interfaccia `java.lang.Comparable` e spiegare come può essere usata per scrivere un metodo di ordinamento di oggetti di tipo non fissato.
   Vedi le API di Java
   `http://java.sun.com/javase/reference/api.jsp`