Accesso alle strutture dinamiche usando gli iteratori

Le strutture dinamiche di dati richiedono costantemente l'esecuzione di cicli per scorrere, uno ad uno, tutti gli elementi dell'insieme.
Per eseguire queste iterazioni si può utilizzare la struttura di controllo *for*. Per esempio, un vettore *poligono* contenente oggetti di classe *Punto* può essere attraversato usando il seguente frammento di codice.

```
for(int i=0; i<poligono.size(); i++)
{
  p = (Punto) poligono.elementAt(i);

  // elaborazioni sull'elemento p
}
```

Java consente di gestire in modo generico i cicli sulle strutture di dati dinamiche. Il meccanismo di gestione delle iterazioni è implementato tramite un oggetto di classe **Iterator**, che può essere ricavato da ogni struttura di dati dinamica tramite il relativo metodo **iterator**.

Per esempio, l'oggetto di classe *Iterator* del vettore *poligono* viene dichiarato nel seguente modo:

```
Iterator i = poligono.iterator();
```

Gli oggetti *Iterator* hanno due metodi principali:

- **next()**, restituisce l'elemento successivo dell'insieme;
- **hasNext()**, restituisce *true* se l'*Iterator* contiene ulteriori elementi.

Il ciclo che scorre tutti gli elementi contenuti nell'*Iterator* è il seguente:

```
while (i.hasNext())
{
  p = (Punto) i.next();

  // elaborazioni sull'elemento p
}
```

L'iterazione precedente, sebbene sia simile a quella implementata con il *for*, ha il vantaggio di essere generica, nel senso che può essere utilizzata con diverse strutture dinamiche e non solo con la classe *Vector*.

Consideriamo per esempio la struttura dinamica **ArrayList**, che è una classe contenuta nel package **java.util** ed è molto simile alla classe *Vector*. Gli oggetti di classe *ArrayList* permettono di memorizzare un insieme variabile di elementi (*array*) e vengono creati usando il costruttore vuoto oppure un costruttore che specifica la capacità iniziale.

```
ArrayList elenco1 = new ArrayList();
ArrayList elenco2 = new ArrayList(10);
```

I metodi principali della classe *ArrayList* che consentono di manipolare l'array sono i seguenti:

- **add**(Object obj), aggiunge un elemento alla fine della lista,
- **remove**(**int** index), toglie dal vettore l'elemento nella posizione indicata dall'indice,
- **size()**, restituisce il numero di oggetti effettivamente memorizzati nell'array,
- **get**(**int** index), restituisce un oggetto di classe *Object* che si trova nella posizione indicata dall'indice,
- **iterator()**, restituisce un oggetto di classe *Iterator*.

Se costruiamo l'oggetto *poligono* con la classe *ArrayList*,

```
ArrayList poligono = new ArrayList();
```

il ciclo per scorrere i suoi elementi può essere implementato con la struttura *for* nel seguente modo:

```
for(int i=0; i<poligono.size(); i++)
{
  p = (Punto) poligono.get(i);

  // elaborazioni sull'elemento p
}
```

Si noti che è stato usato il metodo *get* per leggere un elemento dell'array, invece del metodo *elementAt* utilizzato con la classe *Vector*.

Se avessimo utilizzato un iteratore, il ciclo di scansione della classe *ArrayList* sarebbe stato implementato nello stesso modo con cui è stato implementato per la classe *Vector*.

```
while (i.hasNext())
{
  p = (Punto) i.next();

  // elaborazioni sull'elemento p
}
```