

Gestione dei file ad accesso casuale

I file strutturati possono essere gestiti con un accesso sequenziale, quando si scrivono i dati di seguito a quelli già registrati e si leggono a partire dal primo, oppure con un accesso **casuale** (o **diretto**), quando il programma è in grado di eseguire le operazioni di lettura o scrittura a partire da una posizione scelta.

• Apertura e chiusura

L'apertura di un file ad accesso casuale per le operazioni di input e output, cioè per la lettura e la scrittura, viene eseguita con la dichiarazione di un oggetto di classe **RandomAccessFile**.

```
RandomAccessFile raf = new RandomAccessFile("elenco.dat", "rw");
```

Il costruttore della classe *RandomAccessFile* riceve come primo parametro il nome del file e come secondo parametro la modalità con cui il file deve essere aperto:

- **"r"**: in sola lettura,
- **"rw"**: sia in lettura che in scrittura.

La chiusura di un file ad accesso casuale, sia di input che di output, viene fatta richiamando il metodo *close* nel seguente modo:

```
raf.close();
```

• Posizionamento

I file ad accesso casuale possono essere immaginati come un array di *byte*. Tramite l'utilizzo di un indice viene indicata la posizione da cui iniziano le operazioni di lettura o di scrittura. Dopo l'esecuzione dell'operazione, l'indice viene fatto avanzare del numero di byte letti o scritti.

Il metodo **getFilePointer** restituisce la posizione corrente dell'indice.

Il metodo **seek** posiziona l'indice ad un certo numero di byte calcolati dall'inizio del file.

Per eseguire le scritture in coda al file si deve posizionare l'indice alla fine del file con il seguente comando:

```
raf.seek(raf.length());
```

Il metodo **raf.length()** restituisce la dimensione del file espressa in byte.

• Scrittura

Dopo l'apertura dello stream di output, si possono utilizzare diversi metodi per la scrittura. Ogni metodo memorizza sul file un particolare tipo di dato e assume la forma **writeDato**. Al posto di *Dato* si sostituisce il tipo di dato che si vuole memorizzare.

Per esempio:

```
raf.writeInt(512);  
raf.writeChars("inglese");
```

I valori vengono scritti sul file ad accesso casuale come sequenze di byte a partire dalla posizione dell'indice.

- **Lettura**

Le operazioni di lettura vengono gestite da diversi metodi che assumono la forma **readDato**, dove al posto di *Dato* si sostituisce il tipo di dato che si vuole leggere.

Per esempio:

```
int numero = raf.readInt();  
char c     = raf.readChar();
```

Il primo comando legge quattro byte dal file a partire dalla posizione dell'indice. Il secondo comando legge due byte, corrispondenti alla dimensione del tipo *char*.

Per leggere una porzione di file composta da un numero di byte stabilito a priori si utilizza il metodo **read** e si passa come parametro un array della dimensione opportuna. L'esecuzione del metodo *read* riempie l'array con i dati letti dal file a partire dalla posizione dell'indice.

Per esempio, la lettura di 20 byte a partire dalla posizione 60 viene eseguita con le seguenti istruzioni:

```
byte[] dati = new byte[20];  
raf.seek(60);  
raf.read(dati);
```