Corso di Laurea Ingegneria Informatica

Fondamenti di Informatica 1

Dispensa 11

Array

A. Miola

Dicembre 2007

Contenuti

- Il problema degli studenti da promuovere
- Array
- Array in Java
- Uso di array
  - dichiarazione di variabili array e tipi array
  - creazione di array
  - accesso a un array
  - array e metodi
  - letterali array
- Array e accesso posizionale
  - somma degli elementi di un array
  - visualizzazione degli elementi di un array
  - calcolo del massimo elemento di un array di interi
  - altri esempi ed esercizi

Array

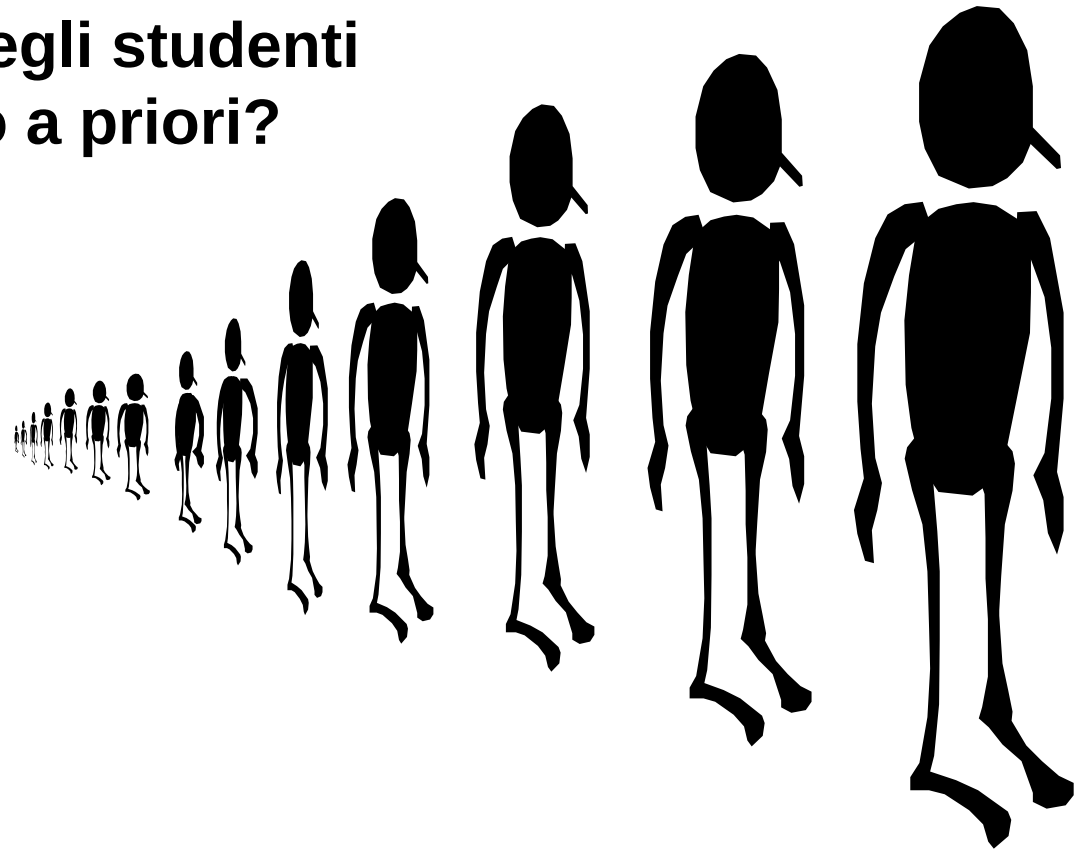
- **Un array è un insieme di variabili omogenee, gestito mediante una singola variabile, a cui è possibile accedere in modo posizionale (ovvero, mediante un indice)**
  - **gli array consentono di gestire dei gruppi di dati**
    - **in quantità prefissata**
    - **che rappresentano informazioni omogenee**
    - **che devono essere elaborati in modo simile**

Il problema degli studenti da promuovere

- **Si consideri il seguente problema**
  - **i venti studenti di un corso hanno svolto una prova di profitto, e ciascuno studente ha ricevuto una votazione in decimi**
    - **le votazioni sono numeri frazionari compresi tra zero e dieci**
  - **il docente del corso ha deciso di promuovere gli studenti la cui votazione è non inferiore ai due terzi del voto medio ottenuto dagli studenti alla prova d'esame**
  - **si vuole scrivere un'applicazione che legge il nome e la votazione di ciascuno dei venti studenti e stampa l'elenco dei nomi degli studenti da promuovere, con le rispettive votazioni**

Limiti della questa prima soluzione

- **Problemi di questa prima soluzione**
  - **se ci fossero 10 studenti?**
  - **se ci fossero 100 o 1000 studenti?**
  - **se il numero degli studenti non fosse noto a priori?**



Caratteristiche del problema degli studenti

- **Questo problema ha due caratteristiche fondamentali – peraltro comuni a molti altri problemi**
  - **vanno gestiti degli insiemi di variabili omogenee**
  - **in ciascun insieme di variabili omogenee, per ciascun elemento devono essere effettuate le stesse operazioni**
- **Vanno gestiti degli insiemi di variabili omogenee**
  - **un insieme di variabili per rappresentare il nome di ciascuno studente**
  - **un insieme di variabili per rappresentare la votazione di ciascuno studente**
  - **ciascuno di questi due insiemi di variabili deve essere composto da un numero di variabili pari al numero degli studenti del corso**

Dagli insiemi di variabili agli array

- **In molti linguaggi di programmazione gli insiemi di variabili sono rappresentati mediante la struttura dati di array**
  - **che cosa è un array?**
  - **che cosa è un array in Java?**
  - **come si usano gli array in Java?**
  - **quali sono le tecniche di programmazione che possono essere usate nella gestione degli array?**

Array

- **Un array è una collezione finita di variabili di uno stesso tipo, posto in corrispondenza biunivoca con un altro insieme finito**
  - **le variabili che compongono un array sono chiamate elementi (o componenti) dell'array**
  - **il numero di elementi di un array è chiamato dimensione dell'array**
  - **gli elementi dell'insieme finito che è in corrispondenza biunivoca con l'array sono chiamati indici**
  - **in un array, ciascun elemento è individuato dal nome dell'array insieme all'indice associato all'elemento**

Array in Java . . .

- **In Java, un array è una sequenza (una collezione ordinata) finita di variabili di uno stesso tipo, posto in corrispondenza biunivoca con un intervallo iniziale e finito dei numeri naturali**
  - **il numero di elementi di un array (la dimensione) è chiamata anche lunghezza dell'array**
  - **l'indice di ciascun elemento in un array è un numero naturale**
    - **l'ordinamento lineare ≤ sui numeri naturali, sulla base dell'indice associato a ciascun elemento, induce un ordinamento sugli elementi di un array**
    - **il primo elemento ha indice 0**
    - **in un array lungo N, l'ultimo elemento ha indice N–1**

. . . Array in Java

- In Java, un array è un oggetto, a cui è possibile accedere mediante un suo riferimento
- L'accesso a un elemento di un array avviene specificando un riferimento all'array seguito dall'indice dell'elemento racchiuso tra parentesi quadre [e]

Dichiarazione di variabili array e tipi array

- **Per usare un array i cui elementi sono di un tipo T bisogna dichiarare una variabile di tipo T[ ]**

```
String[] nomi; // nomi degli studenti
double[] voti; // votazioni ricevute dagli studenti
```

- **Tipi array**
  - **se T è un tipo, l'espressione T[ ] denota un tipo riferimento, chiamato tipo array di elementi di tipo T (array di T)**
  - **gli elementi di un array devono essere tutti di uno stesso tipo**
- **Le variabili di tipo array (o variabili array) sono dunque variabili di tipo riferimento**
  - **la dichiarazione di una variabile array introduce una variabile che può memorizzare il riferimento a un array**
  - **la dichiarazione di una variabile array non associa nessun oggetto array alla variabile dichiarata**

Creazione di array

- **Per creare un array composto da N elementi di tipo T si deve usare l'espressione new T[N]**

```
/* crea l'array per i nomi degli studenti */
nomi = new String[20];
/* crea l'array per i voti degli studenti */
voti = new double[20];
```

- **Gli array sono oggetti**
  - la creazione di un array avviene mediante l'operatore new
  - al momento della creazione di un array, bisogna specificare il tipo e il numero di elementi che compone l'array
  - quando viene creato un array, l'operatore new restituisce il riferimento all'array creato, che può essere memorizzato in una variabile di tipo array

Accesso agli elementi degli array

- **L'operatore [ ] è chiamato operatore di indicizzazione (subscripting)**
  - **è un operatore binario**
    - **il primo operando di [ ] è il riferimento a un array, scritto prima delle parentesi quadre**
    - **il secondo operando di [ ] è una espressione intera che indica l'indice dell'elemento che deve essere acceduto, scritta tra le parentesi quadre**
  - **nell'accesso a un elemento di un array, se il valore dell'indice è minore di zero oppure maggiore o uguale alla lunghezza dell'array, viene causato un errore al tempo di esecuzione di tipo**

    ArrayIndexOutOfBoundsException

    ovvero, di tipo "indice dell'array fuori dai limiti"

Lunghezza di un array

- **La lunghezza (dimensione) di un array è il numero di elementi dell'array**
  - **la lunghezza di un array viene fissata al momento della sua creazione, e non può essere più cambiata**
  - **è possibile accedere alla lunghezza di un array mediante la variabile length**
    - **length è una variabile d'istanza associata agli oggetti array**
    - **si può accedere alla lunghezza di un array mediante l'espressione *riferimento-array* . length**

Attenzione : non confondere la variabile length degli array con il metodo length() delle stringhe

Letterali array. . .

- Un letterale array è la denotazione di un array di cui sono anche specificati i valori iniziali per gli elementi, esso viene scritto elencando in sequenza i letterali che denotano gli elementi dell'array, racchiusi tra parentesi graffe { e } e separati da virgola ,

- I letterali array sono utili, ad esempio, per effettuare il test di metodi che hanno dei parametri array

. . . Letterali array

```
String[] animali;
animali = new String[]{"leone","tigre","orso"};
```

questo frammento di codice è equivalente al seguente

```
String[] animali;
animali = new String[3];
animali[0] = "leone";
animali[1] = "tigre";
animali[2] = "orso";
```

Esercizio

- **Scrivere un metodo int occorrenze(int[ ] a, int v) che calcola il numero di elementi di a che sono uguali a v**
  - **ad esempio, se a è l'array { 1, 3, 2, 3, 5 }**
    - **occorrenze(a,3) deve restituire 2**
    - **occorrenze(a,7) deve restituire 0**

Esercizi . . .

- **Scrivere un metodo int massimo(int[ ] a) che calcola e restituisce il valore del massimo elemento di un array di interi non vuoto a**
  - **Massimo(new int[ ] { 3, 1, 2 }) deve restituire 3**
  - **Massimo(new int[ ] { 3, 7, 5 }) può restituire 7**
- **Scrivere un metodo int posizioneMassimo(int[ ] a) che calcola la posizione dell'elemento di valore massimo di a**
  - **posizioneMassimo(new int[ ] { 3, 1, 2 }) deve restituire 0**
  - **posizioneMassimo(new int[ ] { 3, 1, 3 }) può restituire 0 o 2**

. . . Esercizi . . .

- **Scrivere un metodo boolean crescente(int[ ] a) che verifica se a è ordinato in modo crescente**
  - un array è ordinato in modo crescente se per ogni indice k, l'elemento di indice k è maggiore di tutti gli elementi di indice minore di k
  - è sufficiente verificare se ogni elemento è minore dell'elemento che lo segue immediatamente
  - intuitivamente, bisogna confrontare ciascun elemento di indice i dell'array con l'elemento che lo segue immediatamente, quello di indice i+1
  - attenzione – l'ultimo elemento di un array non ha un elemento che lo segue immediatamente

. . . Esercizi

- **Scrivere un metodo boolean coppiaUguali(int[ ] a) che verifica se a contiene almeno una coppia di elementi uguali**
  - **bisogna confrontare ciascun elemento X dell'array con ogni altro elemento Y dell'array**
    - **dove X e Y sono elementi diversi, nel senso che hanno indici diversi**
  - **bisogna confrontare ciascun elemento di indice i dell'array con ogni altro elemento di indice j dell'array - dove i e j hanno valore diverso**
  - **quindi è necessario utilizzare due variabili indice per accedere agli elementi di uno stesso array**

Esercizi

- Scrivere un metodo int[ ] inverso(int[ ] a) che crea e restituisce un array che contiene gli stessi elementi di a, ma disposti in ordine inverso
- Scrivere un metodo che, dato un array a, calcola e restituisce l'elemento più frequente di a
  - calcolare prima l'array occ delle frequenze degli elementi di a
  - l'elemento da restituire è quello che in a occupa la posizione del massimo di occ

Uguaglianza tra array

- Due array sono uguali se entrambe le seguenti condizioni sono verificate
  - i due array hanno la stessa lunghezza
  - per ciascun indice degli array, gli elementi di indice corrispondente nei due array sono uguali
- Esercizio - scrivere un metodo

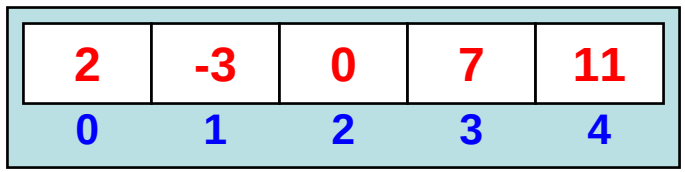
  boolean uguali(String[ ] a, String[ ] b)

  che verifica se gli array di stringhe a e b sono tra loro uguali
- Attenzione
  - in pratica, l'uguaglianza tra array va verificata invocando il metodo di classe boolean equals(a,b) della classe Arrays del package java.util

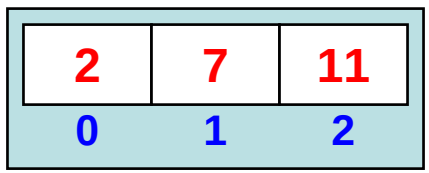
Esercizio

Risolvere il seguente problema

- dato un array A di interi, si vuole creare e calcolare un nuovo array che contiene tutti e soli gli elementi positivi di A, nello stesso ordine in cui occorrono in A (l'array A non deve essere modificato)
- ad esempio, se A è il seguente array



- allora bisogna creare e restituire il seguente array



Esercizio

- **Scrivere un metodo int[ ] cancella(int[ ] a, int p), in cui p indica una posizione valida all'interno di a, che crea e restituisce un nuovo array che contiene tutti gli elementi di a ad eccezione di quello che occupa la posizione p, nello stesso ordine in cui compaiono in a**
  - **ad esempio**

    cancella(new int[ ] { 3, 5, 6, 8 }, 2)

    deve restituire l'array { 3, 5, 8 }

Riferimenti al libro di testo

- Per lo studio di questi argomenti si fa riferimento al libro di testo, e in particolare al capitolo 19 fino al paragrafo 19.5 incluso