

Esercizi sugli Array (parte I)

29 Marzo 2023

Esercizio.

Si definisca una classe di nome `Formica`, che modella oggetti che rappresentano formiche che si muovono in un formicaio (modellato come una matrice quadrata). Un oggetto `Formica` è caratterizzato da un id e da una posizione, definita dagli indici `i` e `j` della cella del formicaio che essa occupa. La classe `Formica` ha infine una variabile di classe `nextId` che rappresenta l'id della prossima formica che sarà creata. La classe `Formica` ha i seguenti costruttori e metodi:

```
/* Costruttore: crea una formica con la posizione specificata
assegnandole un id */
public Formica (int i, int j)

/* Restituisce l'id della formica */
public int getId ()

/* Restituisce l'indice i della posizione della formica */
public int getI ()

/* Restituisce l'indice j della posizione della formica */
public int getJ ()

/* Modifica l'indice i della posizione della formica */
public void setI (int i)

/* Modifica l'indice j della posizione della formica */
public void setJ (int j)

/* Restituisce l'ultimo id assegnato ad una formica */
public static int getLastId ()
```

Si definisca inoltre una classe di nome `Formicaio`, caratterizzata da un array contenente al più 10 oggetti `Formica` e da un array bidimensionale che rappresenta lo spazio che il formicaio occupa. La classe `Formicaio` ha i seguenti costruttori e metodi:

```
/* Costruttore: crea un formicaio il cui spazio è una matrice
quadrata di dimensione dim. Ogni cella dello spazio viene
inizializzata con uno 0. */
public Formicaio (int dim)
```

```

/* Aggiunge una formica nella cella (i,j) del formicaio riportando
il suo id. Il metodo restituisce true se la formica è stata aggiunta
correttamente, false se non è possibile aggiungerla. I motivi perché
l'aggiunta potrebbe non essere possibile sono: la cella (i,j) è già
occupata da un'altra formica; la cella (i,j) non è valida */
public boolean aggiungiFormica (int i, int j)

/* Verifica se una cella (i,j) del formicaio è occupata da qualche
formica */
public boolean occupata (int i, int j)

/* Muove la formica avente l'id specificato in una direzione: 1=NORD,
2=OVEST, 3=SUD, 4=EST. Il metodo restituisce true se la formica è
stata spostata correttamente, false se non è possibile spostarla. I
motivi perché lo spostamento potrebbe non essere possibile sono: la
cella di destinazione è già occupata da un'altra formica; la cella
di destinazione non è valida; l'id specificato non è valido */
public boolean muoviFormica (int direzione, int id)

/* Restituisce una rappresentazione testuale dello spazio del
formicaio: per ogni formica è riportato il suo id in corrispondenza
della sua posizione */
public String toString ()

```

Una volta definite le classi Formica e Formicaio, si scriva una classe `ProvaFormicaio` contenente solo il metodo speciale `main`, il quale deve svolgere le seguenti azioni:

- Chiede all'utente la dimensione dello spazio occupato dal formicaio.
- Crea il formicaio e lo stampa.
- Chiede ripetutamente all'utente se vuole aggiungere una formica al formicaio. Finché l'utente risponde sì, il programma chiede all'utente gli indici della posizione in cui inserire la formica e aggiunge una formica. Il programma comunica se l'aggiunta è andata a buon fine oppure no e stampa il formicaio. Si possono aggiungere al più 9 formiche.
- Chiede ripetutamente all'utente se vuole muovere una formica. Finché l'utente risponde sì, il programma chiede all'utente l'id della formica da muovere e la direzione in cui essa si deve muovere. La direzione può essere specificata tramite un intero (1=NORD, 2=OVEST, 3=SUD, 4=EST). Il programma chiede una direzione finché non viene inserito un valore valido. Il programma comunica se la mossa è stata effettuata correttamente oppure no e stampa il formicaio.