

Fondamenti di Informatica

Prova d'esame del 22/1/2015

Regole d'esame:

1. È vietato parlare con altri studenti
2. È vietato consultare appunti, dispense, libri, in qualunque formato
3. È vietato tenere accesi i telefoni cellulari

Esercizio

Ogni oggetto della classe **BattagliaNavale** rappresenta uno schema per il gioco della battaglia navale. La classe ha una variabile di istanza **schema** (di tipo **int[][]**) e una variabile di classe **idNave**. La variabile **schema** rappresenta la griglia su cui sono disposte le navi. Ogni nave è identificata da un codice identificativo (*id*) che è un numero intero positivo. La variabile di classe **idNave** rappresenta il codice identificativo della prossima nave che verrà inserita. Ogni nave inoltre occupa un certo numero di celle consecutive. Il numero di tali celle è la *lunghezza* della nave. Per semplicità si assume che le navi siano tutte disposte in orizzontale (ogni nave pertanto occupa una serie di celle consecutive sulla stessa riga). La griglia di gioco viene gestita secondo le seguenti regole:

- Una cella il cui valore è 0 non è occupata da navi.
- Una nave con $id\ k > 0$ occupa una serie di celle consecutive di una certa riga. Ciascuna di tali celle contiene il valore k se non è mai stata colpita, e il valore $-k$ in caso contrario.

Una nave che è stata colpita su tutte le sue celle è *affondata*.

La tabella 1 mostra un esempio di schema in cui sono disposte 3 navi identificate dai codici 1, 2, e 3 di lunghezza 3, 2 e 2, rispettivamente. La tabella 2 mostra lo stesso schema dopo che la nave 1 e 2 sono state colpite ciascuna una volta (le celle colpite sono evidenziate in grassetto).

0	0	0	0	0	0
0	1	1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
3	3	0	2	2	0
0	0	0	0	0	0

Tabella 1

0	0	0	0	0	0
0	1	-1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0
3	3	0	2	-2	0
0	0	0	0	0	0

Tabella 2

La classe **BattagliaNavale** è dotata dei seguenti metodi:

- **public BattagliaNavale(int n)**: costruttore della classe. Crea uno schema di dimensione $n \times n$
- **public boolean aggiungiNave(int i, int j, int l)**: aggiunge una nave allo schema. I parametri *i* e *j* sono gli indici della prima cella occupata dalla nave, mentre *l* è la lunghezza della nave. La nave aggiunta avrà come id il valore di **idNave** (che dovrà opportunamente essere aggiornato). Il metodo restituisce **true** se la nave è stata aggiunta correttamente, **false** se non è possibile aggiungere la nave. I motivi perché l'aggiunta di una nave potrebbe non essere possibile sono:
 - almeno una delle celle che la nave dovrebbe occupare è già occupata da un'altra nave;
 - la nave esce fuori dai margini dello schema.
- **public int colpo(int i, int j)**: spara un colpo sulla cella (*i,j*). Qualora il colpo vada a segno (cioè colpisca una nave) lo schema viene aggiornato conseguentemente. Il metodo restituisce uno dei seguenti valori:
 - 0 se il colpo è andato a vuoto
 - 1 se il colpo è andato a segno ma la nave non è stata affondata

- 2 se il colpo è andato a segno e la nave è stata affondata
- **public String toString():** restituisce una rappresentazione testuale dello schema di gioco.

Si scriva la classe **BattagliaNavale** ed una classe **ProvaBattagliaNavale** che contiene il solo metodo **main** e che esegue le seguenti azioni:

- Crea uno schema di battaglia navale chiedendone la dimensione all'utente.
- Chiede ripetutamente all'utente se vuole inserire una nave. Ogni volta che l'utente risponde affermativamente gli viene chiesta la posizione e la lunghezza della nave e gli viene comunicato se la nave è stata inserita o meno.
- Quando l'utente ha terminato di inserire le navi gli viene mostrato lo schema creato;
- Chiede ripetutamente all'utente se vuole tentare di colpire una nave. Ogni volta che l'utente risponde affermativamente gli viene chiesta la cella da colpire e gli viene comunicato l'esito del colpo ("Mancato!", "Colpito!", oppure "Colpito e affondato!").
-

Note:

1. Tutti i file creati devono essere salvati nella cartella C:\fi0122 del proprio PC.
2. Indicare il proprio cognome e nome, la propria matricola e il numero del PC su cui si sta lavorando negli appositi spazi in cima a questo foglio ed anche come commento in testa alla definizione delle classi che si scrivono su file.