Ingegneria Del Software

==========

Battaglia Navale In Java

componenti del gruppo:
Cavalin Alberto, Malvestio Mauro, Guarino Renato

Giugno-Luglio 2003

Indice

1) Il progetto

Il nostro gruppo ha realizzato un programma che implementa il famoso gioco della battaglia navale utilizzando il linguaggio Java.

2) Le specifiche

2.1) Regole e restrizioni:

- Il campo di gioco è definito come un sistema di assi cartesiani, del quale viene utilizzato solo il primo quadrante. In particolare: $x,y \in \mathbb{N}-\{0\}$.
- Ogni punto del suddetto campo è identificato da una coppia di coordinate (x,y) gestite da un'apposita classe.
- Le navi sono posizionate solo orizzontalmente e possono essere di dimensioni comprese tra 1 ed 4 .
- Nella flotta di navi c'è sempre e solo una portaerei la quale ha una componente in più rispetto alle altre, situata sopra alla prima componente da sinistra:

  Es.: Portaerei con 4 componenti

  ```
  [*]
  [*][*][*][*]
  ```

  la lunghezza della portaerei può variare da 1 a 4, come quella delle navi normali.
- Le navi stesse e la portaerei non possono sovrapporsi l'una all'altra.
- La flotta è gestita tramite un vettore di navi il cui primo elemento è una portaerei.

2.2) Modalità di gioco

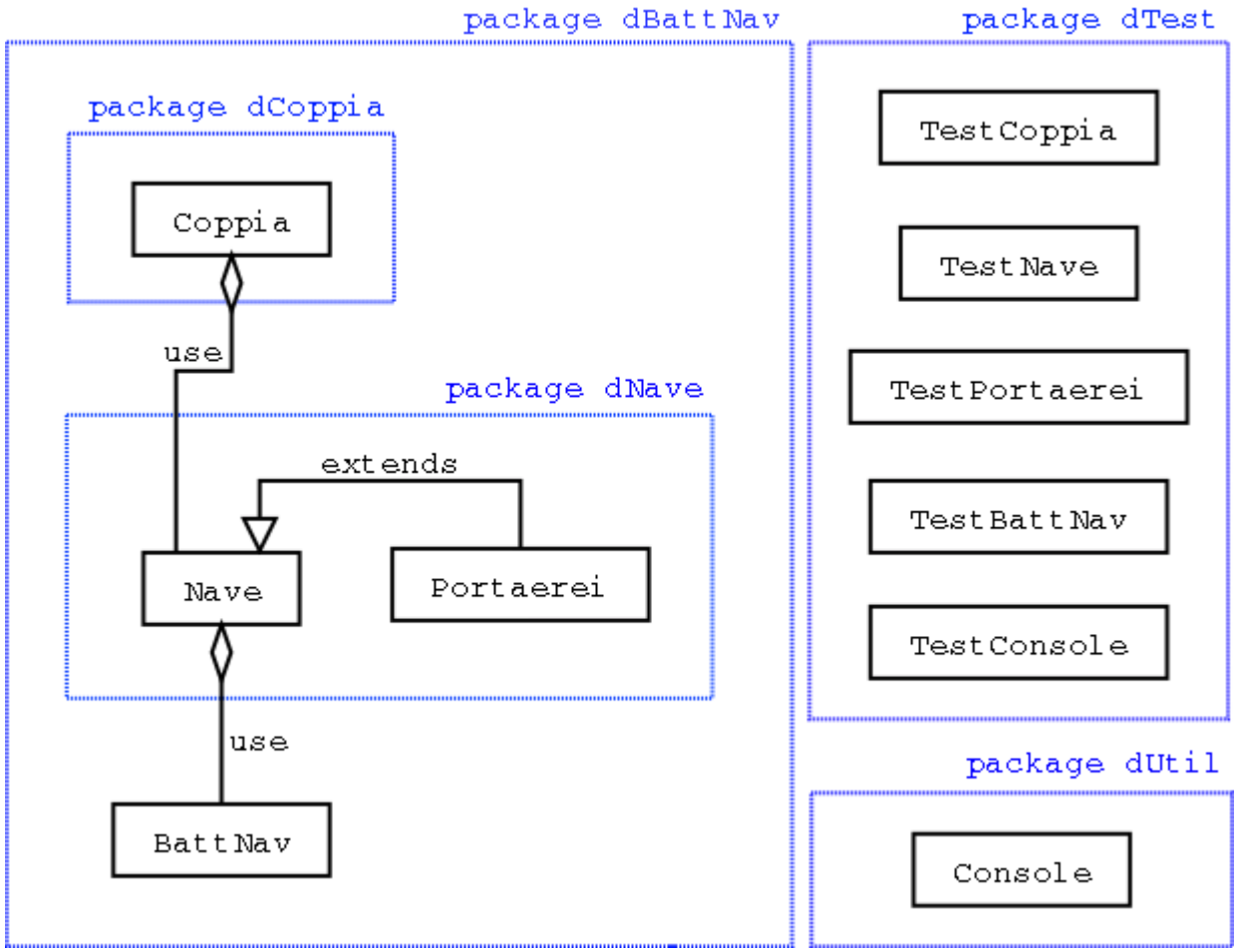
- persona vs sé stessa: il giocatore costruisce un campo di gioco e poi inizia a colpire sullo stesso finché non ha abbattuto tutte le navi
- persona vs persona: i giocatori costruiscono i propri campi, dopo di che iniziano a colpire a turno il campo dell'avversario finché uno dei due non ha eliminato la flotta dell'avversario
- persona vs pc: simile alla precedente con la variante che l'avversario è il calcolatore
- Demo: il calcolatore costruisce da solo il campo, effettuando i dovuti controlli, dopo di che inizia a colpire sullo stesso, tenendo conto dei punti già colpiti, finché non ha abbattuto tutte le navi

3) La struttura

Il nostro gruppo ha scelto di seguire una metodologia top-down per la risoluzione del problema: abbiamo pensato che sarebbe stato molto comodo creare minimo una classe per ogni punto delle specifiche che risultasse più significativo, per poi utilizzarle tutte assieme in un'unica classe che gestisca il tutto senza il minimo sforzo; così, in base a queste considerazioni, abbiamo sviluppato le seguenti classi:

Nome classe	Descrizione
Coppia	gestione di una coppia di coordinate nel campo
Nave	gestione di una nave semplice nel campo
Portaerei	gestione di una nave portaerei nel campo
BattNav	gestione della flotta di navi nel campo

Abbiamo pensato inoltre di inserire le classi in appositi package per sottolinearne le caratteristiche affini; lo schema seguente illustra come il tutto è stato organizzato:



come è possibile notare, sono presenti altri due package: uno per il testing di ogni classe che abbiamo creato ed uno per la gestione ottimizzata dell'input/output da tastiera.

4) Le classi

4.1) Coppia

Questa classe si occupa della gestione di una coppia di coordinate nel campo.

Attributi	
private int x	coordinata x del punto
private int y	coordinata y del punto

Metodi	
public Coppia(int X,int Y)	costruttore di inizializzazione
public int getX()	restituisce la coordinata x del punto
public int getY()	restituisce la coordinata y del punto
public void setX(int X)	imposta la coordinata x del punto
public void setY(int Y)	imposta la coordinata y del punto

4.2) Nave

Questa classe si occupa della gestione di una nave nel campo.

Attributi	
private Coppia coord	coordinate della prima componente della nave
private boolean status[]	stato delle componenti della nave (true=colpito)

Metodi	
public Nave(Coppia c,int l)	costruttore di inizializzazione
public String fuoco(Coppia c)	fuoco ritorna l'esito di un colpo
public String toString()	restituisce stato della nave
public boolean controllo()	controlla se la nave è affondata
public int getX()	restituisce la coordinata x della prima componente
public int getY()	restituisce la coordinata y della prima componente
public int getL()	restituisce la lunghezza della nave
public Coppia getCoords()	restituisce le coordinate della componente iniziale
public boolean [] getStatus()	restituisce il vettore degli stati delle componenti

4.3) Portaerei

Questa classe si occupa della gestione di una portaerei nel campo.

Attributi	
private boolean pista	componente aggiuntiva della nave

Metodi	
public Portaerei(Coppia c,int l)	costruttore di inizializzazione
public String fuoco(Coppia c)	fuoco ritorna l'esito di un colpo
public String toString()	restituisce stato della nave
public boolean controllo()	controlla se la portaerei è affondata
public String toString()	restituisce stato della nave
public int getL()	restituisce la lunghezza della nave
public Coppia getCoords()	restituisce le coordinate della componente iniziale
public boolean [] getStatus()	restituisce il vettore degli stati delle componenti

4.4) BattNav

Questa classe si occupa della gestione delle navi in un campo.

Attributi	
private Nave navi[]	array delle navi della flotta

Metodi	
public BattNav(Nave array_navi[])	costruttore di inizializzazione
public String fuoco(Coppia c)	fuoco ritorna l'esito di un colpo
public String toString()	restituisce stato delle navi
public StringBuffer [] getCampo()	ritorna il campo stampabile con lo stato delle navi
public void stampaCampo()	stampa il campo con lo stato delle navi

4.5) Console

Questa classe contiene metodi per la lettura di numeri e stringhe dallo standard input.

Attributi	
---	---

Metodi	
public Console(InputStream in)	costruttore di inizializzazione
private static void printPrompt(String prompt)	stampa un prompt sullo schermo senza andare a capo
public static String readLine()	legge una stringa terminata da un NEW-LINE
public static String readLine(String prompt)	legge una stringa terminata da un NEW-LINE stampando un prompt
public static int readInt(String prompt)	legge un intero stampando un prompt
public static int readInt()	legge un intero
public static double readDouble(String prompt)	legge un double stampando un prompt
public static double readDouble()	legge un double
public static void clrscr()	cancella lo schermo
public static void pause()	chiede la pressione di un tasto e cancella lo schermo
public static void printHeader (String title)	stampa un titolo formattato
public static void printSubHeader (String title)	stampa un sotto titolo formattato

4.6) Classi nel package dTest

Tutte queste classi contengono solo il metodo main per effettuare i dovuti test, ad eccezione della classe TestBattNav riportata qui di seguito.

Attributi	
---	---

Metodi	
public static void main(String[] args)	metodo principale di avvio
private static void playGame (int ng,int nn)	svolge le funzioni necessarie per lo svolgimento del gioco
private static Portaerei allocaPortaerei (int ng,int g, String nomi[], Random r)	legge/genera le coordinate della portaerei e la alloca
private static void allocaNavi (int ng,int nn,int g, String nomi[], Random r, Nave array_navi[][])	legge/genera le coordinate delle navi e le alloca
private static String leggiColpo (int ng,int g, String nomi[], BattNav campo[], Random r, boolean m[][])	legge/genera un colpo e lo applica al campo restituendo il risultato
private static boolean checkNavi(Nave[] navi,int n, int xn,int yn,int ln)	controlla che una nave appena generata non si sovrapponga a quelle esistenti
private static boolean checkColpo(int x,int y, boolean[][] m)	controlla che un colpo non venga ripetuto nello stesso punto
private static void printHelp ()	stampa il modo d'uso del programma
private static void printCredits ()	stampa delle info sul gioco