

Inserire nel package **it.unipa.community.nomecognome.prg.n11.esX** le seguenti applicazioni.

### 1) GESTIONE DI UN MUSEO.

Scrivete le classi necessarie per la gestione delle opere d'arte dei musei. Un oggetto della classe **Museo** conterrà nome, indirizzo e l'elenco delle opere d'arte presenti nel museo.

Scrivete una gerarchia per rappresentare i seguenti oggetti: **OperaDArte**, **Scultura**, **Dipinto**. Tutti gli oggetti devono avere un titolo, un autore (String) e la dataDiCreazione (per semplicità un oggetto della classe String) inizializzati dal costruttore. Ogni **Scultura** ha in aggiunta il materiale e l'altezza mentre ogni **Dipinto** ha la tecnica (olio, tempera, litografia, serigrafia, china) e le dimensioni. Tutti i membri devono essere incapsulati.

La classe **Museo** deve avere un metodo per stampare la lista delle opere contenute e metodi per l'inserimento e la cancellazione di opere d'arte.

Utilizzate le vostre classi in un programma che crea un oggetto **Museo** e mostra un menù interattivo testuale in cui si dà all'utente la possibilità di aggiungere nuovi elementi da tastiera, stampare i dettagli di tutti gli oggetti presenti nel museo, eliminare un elemento individuato in base al titolo, stampare i dettagli di un'opera individuata attraverso il titolo.

### 2) SIMULAZIONE TRAFFICO MARITTIMO

Scrivete un programma per la simulazione del traffico marittimo in prossimità di un porto.

Si supponga che ogni nave sia caratterizzata da un codice univoco (String) e abbia una posizione in coordinate cartesiane (double), una velocità (double), una direzione (int fra 0 e 359), lunghezza e larghezza (entrambi double) e numero di passeggeri a bordo (int).

Le navi devono avere un metodo `avanza()`, che aggiorna le coordinate in base alla velocità e alla direzione (vedi suggerimenti).

Scrivete una classe **Nave** con tutti i dati, metodi e costruttori che ritenete opportuni.

Scrivete una classe **Porto** che, attraverso un suo metodo, permette di invocare il metodo `avanza()` di tutti i natanti presenti nello stretto.

La classe **Porto** deve permettere l'inserimento e l'eliminazione dei vari natanti.

Prevedete nella classe **Porto** anche un metodo per la verifica e prevenzione di collisioni (per semplicità il messaggio di pericolo verrà attivato quando la distanza fra due natanti è inferiore a una certa soglia che fisserete voi arbitrariamente).

Utilizzate le classi di cui sopra in un programma che crea un oggetto **Porto**, gestisce l'interazione con l'utente mediante un menu per inserimento o eliminazione di navi e, quando richiesto, invoca il metodo che fa avanzare tutti i natanti, stampa lo stato di tutte le navi e verifica la presenza di potenziali collisioni.

#### Suggerimenti:

La nuova posizione della nave si calcola con le formule:

```
x+=velocita*cos(direzione*3.14/180);  
y+=velocita*sin(direzione*3.14/180);
```

La distanza tra due punti P0(x0, y0) e P1(x1, y1) si può calcolare nel seguente modo:

```
double dx = x1 - x0;  
double dy = y1 - y0;  
double distanza = sqrt(dx*dx + dy*dy);
```