

Nome e cognome: _____ Matricola: _____

Programmazione ad oggetti mod.1, esame del 30/06/2023

PROVA DI PROGRAMMAZIONE

Durante questa prova si possono utilizzare il libro di testo, appunti o altro materiale (come, ad esempio, la stampa delle slide del corso). È invece proibito l'utilizzo di dispositivi elettronici quali, ad esempio, computer portatili, smartphone ed e-reader. Le risposte devono essere date scrivendo il codice sotto il testo dell'esercizio e, se necessario, sul retro del foglio. La durata della prova è di 60 minuti.

Nello svolgimento degli esercizi:

- È necessario massimizzare sempre l'information hiding,
- Le classi possono implementare metodi aggiuntivi se ritenuto utile o necessario, ma questi non devono essere visibili al di fuori della classe stessa.

Esercizio 1 (5 punti): Submarine

Si implementi una classe Submarine che rappresenta un sottomarino. In particolare, tale classe deve memorizzare il nome del sottomarino, la profondità massima a cui può andare il sottomarino, e la profondità attuale. Il livello di profondità è espresso in metri con un valore numerico floating point. Tali informazioni devono essere inizializzate durante la costruzione dell'oggetto, devono essere esternamente accessibili ma non modificabili.

Si implementi poi un metodo immersione che riceve un valore floating point che rappresenta di quanto il sottomarino si deve immergere. Nel caso in cui tale valore sia compatibile con la profondità massima, questo metodo deve aggiornare la profondità coerentemente e ritornare un valore true, o in caso contrario non deve modificare la profondità ritornando false.

Esercizio 2 (7 punti): PoweredSubmarine

Si estenda la classe Submarine con una classe PoweredSubmarine che rappresenta una sottomarino motorizzato. In particolare, il PoweredSubmarine contiene un valore numerico floating point che rappresenta di quanto il sottomarino si immerge quando il motore è acceso, e un flag booleano che è true se il motore è acceso. Si implementino poi due metodi pubblici per accendere e spegnere il motore, e un metodo step() che se il motore è acceso lo fa scendere della profondità attesa, e che ritorna true se e solo se alla fine del metodo il sottomarino è sceso della profondità attesa.

Esercizio 3 (3 punti): rescueSubmarine

Si implementi un metodo statico `rescueSubmarine` che riceve una collezione di `Submarine`. Per ciascun sottomarino all'interno della collezione, controlla se la profondità attuale del sottomarino è maggiore di quella massima, e in tal caso crea un `PoweredSubmarine` che può raggiungere tale profondità con il motore acceso.