



Esercizio 1 [80 punti]

Si realizzi un programma per la gestione degli stand enogastronomici di una piccola sagra paesana. Per la gestione delle ordinazioni è presente un totem con schermo touch. Il cliente potrà inserire le ordinazioni e pagare con carta di credito o simili. Le ordinazioni verranno poi trasmesse nella zona di preparazione ed evase in ordine d'arrivo. Per esigenze di semplificazione il cliente potrà ordinare un solo menu tra i tre menu decisi a priori (menu A, B o C).

Realizzare la **classe Menu** descritta di seguito:

attributi:

- *nome* : stringa che identifica la tipologia di menu
- *costo* : double che determina il costo di un menu
- *descrizione* : stringa contenente la descrizione del menu

metodi:

- *Menu(3 parametri)* : istanzia l'oggetto a partire dal nome, dal costo e dalla descrizione
- *GetNome()* : restituisce il nome. Sostituibile con una **proprietà**.
- *GetCosto()* : restituisce il costo. Sostituibile con una **proprietà**.
- *GetDescrizione()* : restituisce la descrizione. Sostituibile con una **proprietà**.

Realizzare la **classe Ordinazione** descritta di seguito:

attributi:

- *ID* : intero che identifica l'id d'ordine (un numero progressivo)
- *menuScelto* : oggetto di tipo Menu che identifica il menu scelto
- *erogato* : variabile di tipo DateTime che fissa l'ora di erogazione
- *evaso* : booleano che determina se l'ordine è ancora in lavorazione o meno

metodi:

- *Ordinazione(2 parametri)* : istanzia l'oggetto specificando l'id e il menu scelto
- *getID()* : restituisce l'ID. Sostituibile con una **proprietà**.
- *getMenuScelto()* : restituisce il menu scelto. Sostituibile con una **proprietà**.
- *getEvaso()* : restituisce l'attributo evaso. Sostituibile con una **proprietà**.
- *TempoDiAttesa()* : restituisce il tempo di attesa dell'ordinazione (in minuti). Sostituibile con una **proprietà**.
- *Evadi()* : setta l'attributo *evaso* a true. Sostituibile con una **proprietà**.
- *ToString()* : override del metodo ToString() della classe Object che restituisce sotto forma di stringa lo stato interno dell'oggetto.

Realizzare la classe **GestoreOrdinazioni** descritta di seguito:

attributi:

- *ordinazioni* : lista di oggetti Ordinazione

metodi:

- *GestoreOrdinazioni()* : costruttore di default, nessun parametro
- *TotaleOrdinazioni()* : restituisce il numero di ordinazioni totali. Sostituibile con una **proprietà**.
- *TutteLeOrdinazioni()* : restituisce la **lista** di tutte le ordinazioni presenti.
- *AggiungiOrdinazione(1 parametro)* : inserisce una nuova ordinazione, restituisce un booleano in base all'esito dell'operazione



- *StampaOrdinazione(1 parametro)* : cerca l'ordinazione con **id** passato come parametro e restituisce una stringa contenente lo stato interno dell'oggetto o una stringa vuota se l'ordinazione non è presente nella lista
- *RimuoviOrdinazione(1 parametro)* : rimuove l'ordinazione con **id** passato come parametro (se **non** evasa), restituisce un booleano in base all'esito dell'operazione
- *GetTempoAttesa(1 parametro)* : restituisce il tempo di attesa dell'ordinazione con **id** passato come parametro (se **non** evasa), altrimenti restituisce -1
- *EvadiOrdinazione(1 parametro)* : evade l'ordinazione con **id** passato come parametro (se **non** evasa), restituisce un booleano in base all'esito dell'operazione
- *IncassoTotale()* : restituisce l'incasso totale basandosi sul costo dei menu. Sostituibile con una **proprietà**.
- *GetNonEvasa()* : restituisce il **numero** di ordinazioni ancora da evadere. Sostituibile con una **proprietà**.

Esercizio 2 [10 punti]

Si crei un'applicazione consumer di tipo console atta a testare le tre classi appena create.

Best practices: [10 punti]

Uso di nomi variabili opportuni, presenza di commenti, codice correttamente indentato e senza errori di compilazione.

NOTE:

Realizzare le classi e il relativo consumer per il test nello **stesso file**.

Consegnare su Moodle soltanto il file .cs rinominato in **<cognome-data-classe.cs>**

Tempo massimo: 50 minuti.

Il voto sarà determinato nel seguente modo: **punteggio conseguito / 10**