

Traccia2

Una società vuole sviluppare una applicazione per la gestione di ordini di prodotti da parte dei propri clienti. Lo scopo ultimo è di visualizzare il riepilogo dell'ordine di un cliente che specifichi per ciascun prodotto presente nell'ordine alcuni dati, e l'importo totale (si pensi ad esempio al riepilogo dell'ordine mostrato prima del suo definitivo invio in Amazon). Si sviluppi una classe per rappresentare i clienti. La classe ha le variabili membro codice (stringa C allocata staticamente) e indirizzo (di tipo string) e fornisce la funzione get per il codice. Il codice deve consentire di riconoscere i soggetti con partita IVA dai soggetti senza partita IVA, per fissare le idee si utilizzi a tale scopo il primo carattere del codice.

Un prodotto può essere un bene oppure un servizio. Un prodotto ha un identificativo del prodotto (sia una stringa C allocata staticamente) un genere (che tipo di prodotto è, sia una stringa C allocata dinamicamente) e un costo unitario (sia un numero reale). Un bene specifica inoltre se è un bene di prima necessità (informazione booleana) e una descrizione (stringa C allocata dinamicamente). Un servizio specifica invece il tipo di servizio (stringa C allocata dinamicamente). Si implementino le relative classi. Ognuna di esse deve fornire le funzionalità necessarie per una corretta gestione dell'estensione dinamica e una funzione per il calcolo dell'imposta (IVA) che riceve in ingresso una stringa rappresentante il codice cliente e restituisce il valore che deve essere sommato al costo del prodotto per ottenere il prezzo totale. La funzione calcola tale valore come spiegato in seguito. La sola classe prodotto fornisce inoltre le funzioni membro get.

Il valore di imposta è dovuto dai soggetti senza partita IVA. Quindi per i soggetti con partita IVA la funzione fornirà sempre valore 0. Per gli altri, nel caso in cui si tratti di un bene di prima necessità l'imposta è il 4% del costo (aliquota minima), altrimenti è il 22% (aliquota ordinaria). Nel caso di servizi l'imposta è il 10% (aliquota ridotta). Il valore tornato dalla funzione dovrà essere aggiunto quindi al prezzo del prodotto nel momento in cui si deve calcolare il suo costo effettivo.

Si sviluppi quindi una classe che rappresenti ordini. Tale classe deve consentire di risalire al cliente che ha effettuato l'ordine attraverso una opportuna variabile membro, e di memorizzare i prodotti ordinati attraverso un opportuno vettore di record allocato staticamente e il relativo riempimento. Il record deve avere un campo che consente di risalire al prodotto, un campo che memorizza la quantità ordinata e un campo per il costo effettivo relativo a quella quantità. La classe ordine deve consentire di aggiungere un elemento al vettore (nella prima posizione libera) e di calcolare il totale dell'ordine. La funzione di inserimento lancia una eccezione nel caso si tenti di inserire un record se il vettore è pieno.

Si sviluppi infine un programma di test per le classi realizzate. Per semplicità è il programma di test che realizza l'opportuno collegamento tra un cliente e il/i suoi ordini e aggiunge elementi agli ordini.

Di seguito un esempio di esecuzione:

prodotto P1 aggiunto all'ordine

prodotto P2 aggiunto all'ordine

Eccezione: impossibile aggiungere un ulteriore prodotto all'ordine

Dettaglio Ordine

Codice Cliente: A111

Prodotto: servizio di catering Quantita: 1

Costo 2500

Prodotto: dvd Quantita: 1

Costo 52.46

Totale ordine: 2552.46