

Esercizio 11.15.

Si consideri la seguente realtà. Un supermercato prevede la possibilità per i clienti di dotarsi di fidelity card e diventare quindi clienti “fedeli” al supermercato. La fidelity card permette di partecipare alla raccolta punti cosicché al raggiungimento di una certa soglia punti i clienti fedeli vinceranno un premio. Inoltre i clienti fedeli godono di uno sconto incondizionato del 5% sulla spesa. Si chiede di modellare tale realtà aderendo alle seguenti specifiche.

1. Definire una classe `Prodotto` i cui oggetti rappresentano un prodotto in vendita nel supermercato. Un prodotto è quindi caratterizzato dal suo prezzo di vendita.
2. Definire una classe `Cliente` i cui oggetti rappresentano un generico cliente del supermercato. Un cliente è caratterizzato dal “carrello della spesa”, cioè dalla lista dei prodotti acquistati nel corso di una spesa, rappresentabile tramite un vettore di oggetti `Prodotto`.
 - (a) La classe `Cliente` rende disponibile un metodo `double spesaTotale()` con il seguente comportamento: `c.spesaTotale()` ritorna il prezzo totale da pagare per il carrello della spesa del cliente `c`.
3. Definire una classe `ClienteFedele` derivata da `Cliente` i cui oggetti rappresentano un cliente fedele, ovvero dotato di fidelity card. Ogni cliente fedele è caratterizzato dal saldo dei punti accumulati. D'altra parte la soglia punti `SP` per ottenere la vincita del premio è fissata per tutti i clienti fedeli a 100, mentre lo sconto incondizionato `Sc` è fissato per tutti i clienti fedeli al 5%.
 - (a) Il comportamento di `spesaTotale()` per un cliente fedele è aggiornato nel seguente modo: `cf.spesaTotale()` ritorna il prezzo totale da pagare per il carrello della spesa del cliente `cf` dove ogni prodotto nel carrello della spesa gode dello sconto incondizionato `Sc`.
 - (b) La classe `ClienteFedele` fornisce un metodo `void accreditaPunti(int)` con il seguente comportamento: `cf.accreditaPunti(n)` incrementa di `n` il saldo punti del cliente fedele `cf`; inoltre se dopo l'aggiornamento il saldo punti è maggiore o uguale alla soglia punti `SP` per ottenere il premio, decrementa il saldo punti di `SP` e quindi lancia una eccezione di tipo `RitiroPremio`; si chiede anche di definire tale classe `RitiroPremio`.
4. Definire una classe `GestioneGiornaliera` i cui oggetti permettono la gestione giornaliera della cassa e dei punti fedeltà del supermercato. Un oggetto della classe `GestioneGiornaliera` è caratterizzato dalla lista giornaliera dei clienti del supermercato, rappresentabile tramite un vettore.
 - (a) `GestioneGiornaliera` fornisce un metodo `double chiudiCassa()` con il seguente comportamento: `g.chiudiCassa()` deve ritornare l'incasso giornaliero totale del supermercato, ovvero per la lista giornaliera dei clienti rappresentata da `g`, ed accreditare a tutti i clienti fedeli della giornata i punti fedeltà accumulati secondo la seguente regola: per un cliente fedele con una spesa totale di `s` euro vengono accreditati `s/10` punti (dove `s/10` denota il troncamento intero della divisione).
 - (b) `GestioneGiornaliera` fornisce un metodo `int saldoPuntiGiornaliero()` con il seguente comportamento: `g.saldoPuntiGiornaliero()` ritorna il totale del saldo punti per tutti i clienti fedeli della lista giornaliera dei clienti rappresentata da `g`.