

# RELAZIONE PROGETTO DISTRIBUTORE

Il progetto implementa un sistema software per la gestione di un distributore automatico di prodotti alimentari e bevande. Il sistema permette agli utenti di selezionare prodotti, effettuare pagamenti sia in contanti che con carta, e ricevere il resto quando necessario.

## Architettura del Sistema

### Classi Principali

1. **Prodotto**: Classe base astratta che rappresenta un generico prodotto nel distributore.
  - Attributi: nome, prezzo, tipo (CIBO o BEVANDA)
  - Metodi: getter per gli attributi e toString()
2. **Cibo e Bevanda**: Sottoclassi di Prodotto che specificano il tipo di prodotto.
  - Estendono Prodotto con il tipo specifico (CIBO o BEVANDA)
3. **Carta**: Rappresenta una carta di pagamento con saldo.
  - Attributi: intestatario, numero carta, saldo (inizializzato a 250€)
  - Metodi: paga(importo), getSaldo(), toString()
4. **Distributore**: Classe principale che gestisce la logica del distributore automatico.
  - Implementa l'interfaccia DistributoreInterface
  - Gestisce i prodotti organizzati per file
  - Gestisce il pagamento sia in contanti che con carta
  - Calcola e restituisce il resto

5. **DistributoreInterface**: Interfaccia che definisce i metodi principali del distributore.

- mostraProdotti()
- sceltaProdotto()
- aggiungiProdotto()

6. **MainDistributore**: Classe principale con il metodo main per l'avvio del programma.

Enum

- **TipoProdotto**: Enum che definisce i tipi di prodotti disponibili (CIBO, BEVANDA)

## VANTAGGI

La classe `Distributore` (che implementa l'interfaccia) può modificare i dettagli interni (es. logica di pagamento, struttura dati per i prodotti) **senza cambiare** il modo in cui altre classi (es. `MainDistributore`) interagiscono con essa.

Se domani volessimo creare un `DistributorePremium` con funzionalità aggiuntive (es. sconti per utenti registrati), potremmo farlo **implementando la stessa interfaccia**, garantendo compatibilità con il codice esistente.

Senza interfaccia, saremmo costretti a modificare tutti i punti del codice dove il `Distributore` viene usato.

**ENUM** (enumerazione) è un tipo di dato in Java che **definisce una lista fissa di valori costanti**. Abbiamo usato `enum` per **limitare i valori possibili** (solo `CIBO` o `BEVANDA`) ed **evitare errori** nel codice. È più sicuro e leggibile di usare stringhe o numeri.

## Funzionalità Principali

### 1. Gestione Prodotti:

- I prodotti sono organizzati in file numerate (da 1 a 4)
- Possibilità di aggiungere nuovi prodotti alle file
- Visualizzazione completa dei prodotti disponibili

### 2. Selezione Prodotti:

- L'utente può selezionare un prodotto specificando fila e indice
- Visualizzazione del prodotto selezionato con prezzo

### 3. Sistema di Pagamento:

- **Pagamento in contanti:**
  - Accetta monete e banconote (da 0.05€ a 20€)
  - Calcolo automatico del saldo inserito
  - Restituzione del resto con il minor numero di monete possibili
- **Pagamento con carta:**
  - Verifica del saldo disponibile
  - Aggiornamento del saldo della carta
  - Messaggi di conferma o rifiuto del pagamento

### 4. Gestione Errori:

- Controllo della validità della fila selezionata
- Controllo della validità dell'indice del prodotto
- Verifica monete valide durante l'inserimento
- Gestione del saldo insufficiente

## ESECUZIONE DEL PROGRAMMA

1. Inizializzazione del distributore con prodotti predefiniti
2. Visualizzazione del menu prodotti
3. Selezione da parte dell'utente di fila e prodotto
4. Scelta del metodo di pagamento (contanti o carta)
5. Esecuzione del pagamento:
  - Per contanti: inserimento monete fino a raggiungere/superare il prezzo
  - Per carta: verifica immediata del saldo
6. Conferma acquisto o messaggio di errore
7. Eventuale restituzione del resto
8. Possibilità di effettuare un nuovo acquisto o uscire

## POSSIBILI MIGLIORAMENTI

1. Aggiunta di un sistema di autenticazione per le carte
2. Aggiunta di un sistema di gestione inventario (quantità prodotti)
3. Implementazione di un'interfaccia grafica
4. Estensione dei tipi di prodotto (es. dolci, salati, etc.)

Potremmo implementare un'interfaccia che all'avvio del programma faccia scegliere se entrare nella sezione addetto (con un codice di verifica) che ha il potere di :

1. Inserire nuovi prodotti
2. Ritirare il denaro
3. Modificare i prezzi dei prodotti

o faccia entrare nella sezione cliente cioè quella progettata da noi.