**PROGRAMMAZIONE III**

**Progetto “Distributore”**

Relazione

**Partecipanti**

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Giuseppe | Francesco | Salvatore |
| Cognome | Franzese | D’Angelo | Iovinella |
| Matricola | 0124001941 | 0124001792 | 0124001927 |

Sommario

[Descrizione del progetto 2](#_Toc65258666)

[Design pattern 3](#_Toc65258667)

[Memento 4](#_Toc65258668)

[State 4](#_Toc65258669)

[DAO 5](#_Toc65258670)

[Diagramma delle classi 6](#_Toc65258671)

[Documentazione 7](#_Toc65258672)

[Creazione tabelle 7](#_Toc65258673)

[Credenziali ADMIN 7](#_Toc65258674)

[Connessione al Database su Java 7](#_Toc65258675)

[Screen utili 8](#_Toc65258676)

[Menù iniziale 8](#_Toc65258677)

[Menù login 8](#_Toc65258678)

[Menù amministratore 9](#_Toc65258679)

[Menù cliente 9](#_Toc65258680)

[Lista prodotti/categorie 10](#_Toc65258681)

# Descrizione del progetto

Si vuole realizzare e sviluppare un distributore generico di prodotti. Il distributore contiene prodotti che sono divisi in categorie. Si vogliono monitorare i dati persistenti del distributore mediante un database, implementato con Apache Derby. Tutti i passaggi riguardanti la creazione delle tabelle verranno poi descritti nella documentazione nelle pagine successive. I dati da rendere persistenti sono quelli relativi ad ogni singolo prodotto: codice, nome, prezzo, quantità e categoria d’appartenenza; e i dati relativi ad ogni singola categoria: codice, nome. L’accesso al sistema può avvenire secondo due modalità: Amministratore e Cliente.

L’amministratore può eseguire le seguenti operazioni:

* Aggiungere, rimuovere, modificare una categoria;
* Aggiungere, rimuover, modificare un prodotto;
* Definire il prezzo per ogni prodotto;
* Visualizzare la cronologia di tutti gli acquisti effettuati;

Il sistema inoltre controlla se la quantità di un determinato prodotto scende al di sotto della quantità minima, che è 2, e provvederà a inviare una notifica all’amministratore, il quale, alla ricezione, provvederà ad aggiornare la quantità del relativo prodotto.

Il cliente può eseguire le seguenti operazioni:

* Scegliere un prodotto;
* Effettuare il pagamento, il quale avviene secondo due modalità:

1. Contanti;
2. Carta di credito;

Il cliente può effettuare l’acquisto di più prodotti contemporaneamente, quindi può scegliere più prodotti insieme e poi effettuare un unico pagamento. Il distributore, una volta che il cliente termina la fase di scelta del prodotto, mostrerà l’importo totale del pagamento e una finestra in cui il cliente potrà scegliere la modalità di pagamento desiderata.

Nel caso in cui il cliente scelga di pagare in contanti, ad ogni inserimento di una moneta o di una banconota, il sistema provvederà a calcolare l’importo rimanente, e lo mostrerà al cliente. In caso in cui il cliente inserisca un importo superiore a quello previsto, il sistema provvederà a calcolare, mostrare ed erogare il resto. I contanti accettati sono:

* 0,10 – 0,20 – 0,50 centesimi;
* 1 – 2 euro;
* 5 – 10 – 20 euro;

Dopo aver inserito i contanti richiesti per effettuare il pagamento, il sistema provvederà ad erogare il prodotto o i prodotti selezionati dal cliente.

Nel caso in cui il cliente scelga di pagare con carta di credito, una volta che quest’ultimo ha inserito la carta, il sistema provvederà a mostrare una finestra in cui il cliente potrà inserire il proprio PIN. Una volta verificata la correttezza del PIN, il sistema effettuerà la transazione ed erogherà il prodotto o i prodotti selezionati dal cliente.

# Design pattern

I design pattern utilizzati sono:

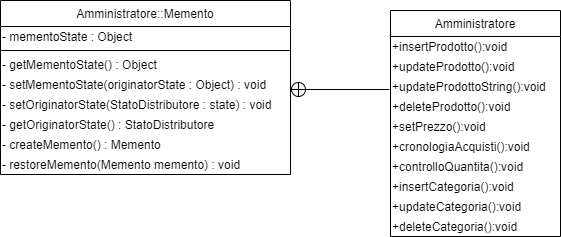
* State: pattern comportamentale che viene utilizzato quando il comportamento di un oggetto deve cambiare in base al suo stato. All’interno del progetto viene utilizzato per monitorare lo stato del distributore che varia tra: In funzione, Guasto. (Lato amministratore).
* Memento: pattern comportamentale che viene utilizzato quando si ha necessità di ripristinare lo stato di un oggetto ad un suo precedente stato. All’interno del progetto viene utilizzato per: in caso di cambio di stato del distributore, ritornare allo stato precedente. Esempio: se il distributore era in attesa e da un momento all’altro passa in uno stato di guasto, il distributore, tornerà allo stato precedente, risolvendo appunto il guasto e tornando in attesa. (Lato amministratore).

Ulteriori strutture utilizzate:

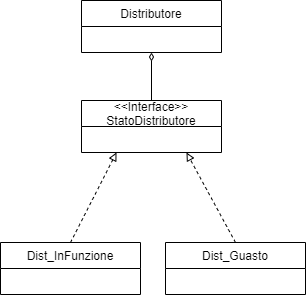
* DAO: pattern architetturale per la gestione della persistenza.

Di seguito vengono riportate alcune immagini che indicano come i pattern sono stati inseriti all’interno del diagramma delle classi:

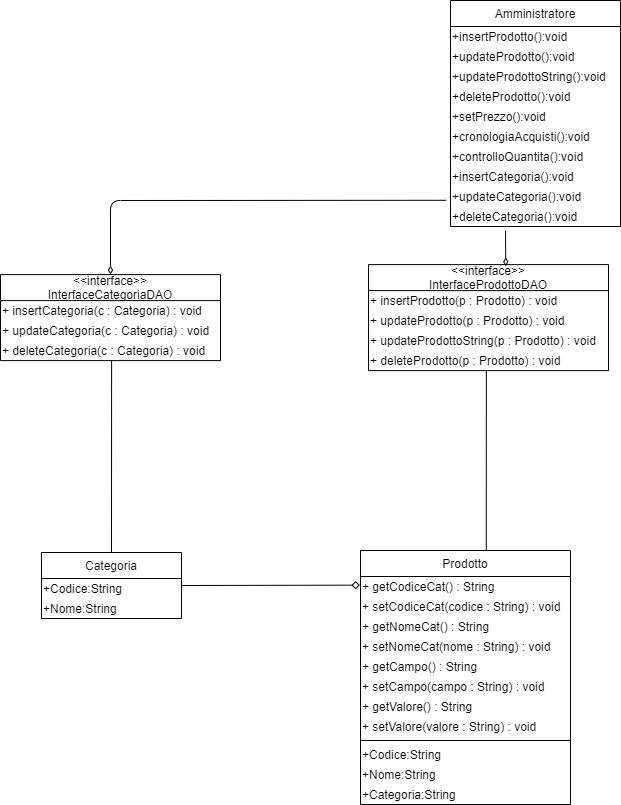
## Memento



## State

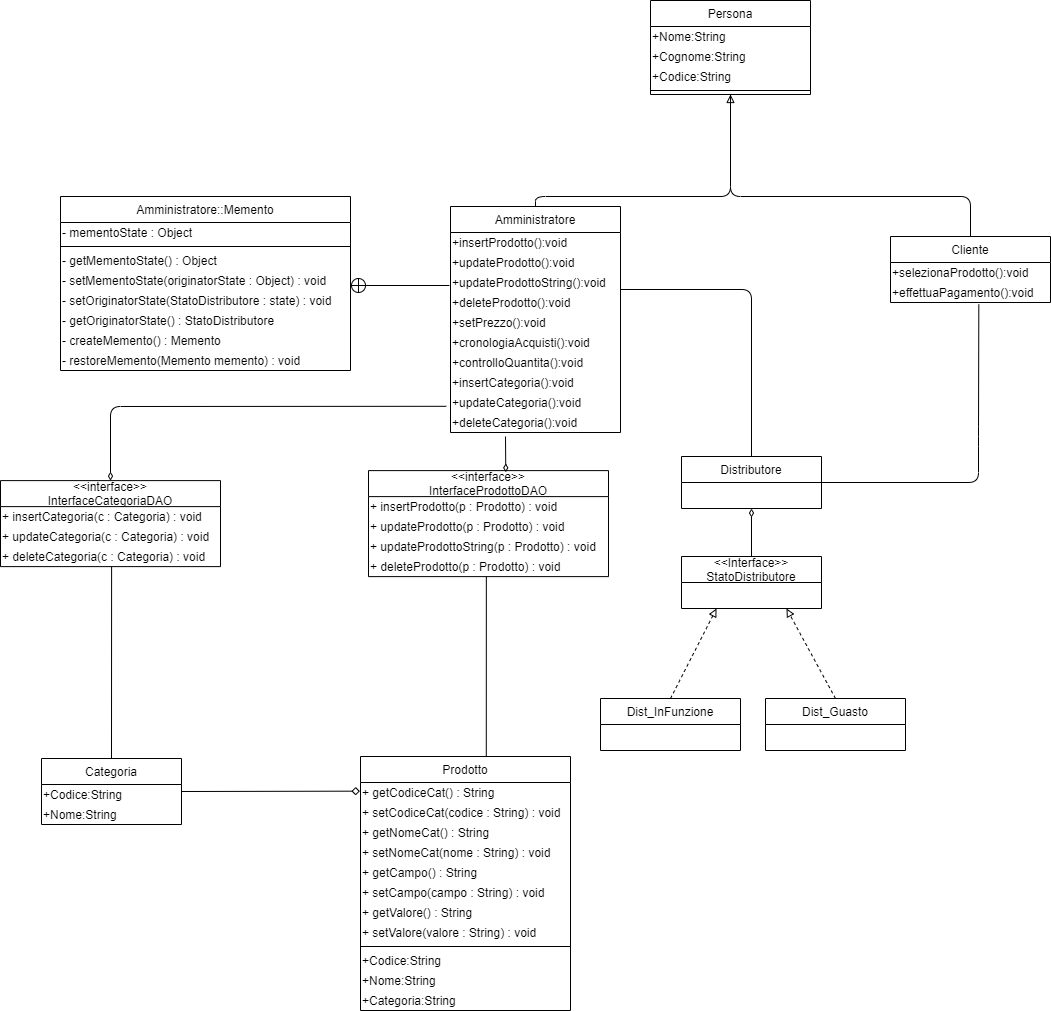


## DAO



# Diagramma delle classi

Di seguito viene riportato il diagramma delle classi sviluppato durante la fase di progettazione:



# Documentazione

## Creazione tabelle

Le tabelle create per memorizzare i vari dati nel Database sono principalmente tre:

1. **Prodotto:** Permette la memorizzazione di tutti i prodotti nonché la modifica e l’eliminazione degli stessi;
2. **Categoria:** Permette la memorizzazione di tutte le categorie nonché la modifica e l’eliminazione delle stesse;
3. **Cronologia:** Memorizza tutti i prodotti acquistati dai vari clienti per poter poi effettuare uno storico di questi acquisti.

Le query per la creazione delle varie tabelle sono:

1. **Prodotto:** create table prodotto (codice varchar (25) not null, nome varchar (25) not null, prezzo double not null, quantita double not null, categoria varchar(25) not null, primary key(codice),foreign key(categoria) references categoria(codice));

2. **Categoria:** create table categoria (codice varchar (25) not null, nome varchar (25) not null, primary key(codice));

3. **Cronologia:** create table cronologia (id varchar (10));

## Credenziali ADMIN

Le credenziali dell’ADMIN per accede al relativo menù sono:

* Nome: admin
* Cognome: admin
* ID: 01

## Connessione al Database su Java

Con i seguenti comandi viene effettuata la connessione al database che nel nostro caso è Derby:

final String jdbcURL = "jdbc:derby:distributore;create=true";

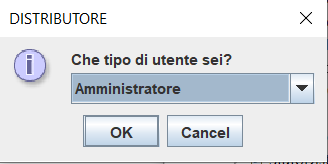
Connection connection;  
        Statement statement;  
        connection = DriverManager.getConnection(jdbcURL);

# Screen utili

In questa sezione vengono riportati tutti gli screen utili a capire il funzionamento del nostro distributore:

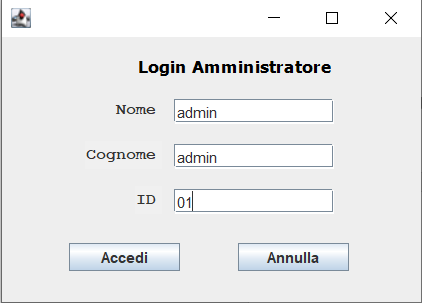
## Menù iniziale

Menù iniziale mostrato all’ avvio del nostro programma:



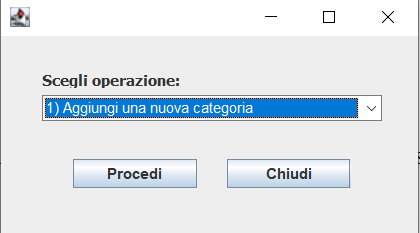
## Menù login

In questa sezione vanno inseriti i dati per accede alle funzionalità

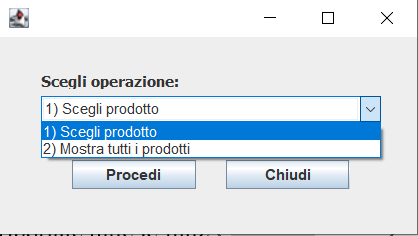
dell’amministratore:

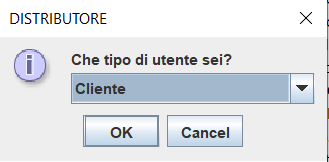
## Menù amministratore

Immagine che contiene testo

Descrizione generata automaticamenteIn questo menù vengono riportate tutte le funzionalità che l’amministratore può svolgere:

## Menù cliente

Di seguito vengono riportate tutte le funzionalità che il cliente può svolgere:



## Lista prodotti/categorie

Di seguito vengono riportate la lista delle categorie e dei prodotti disponibili nel distributore:



