



# Università degli Studi di Salerno

Corso di Ingegneria del Software



**Negozio-Online**

**ODD – Object Design Document**

**Versione 1.3**

**Partecipanti:**

Nome	Matricola
Gaglione Rino	0512102476
Capobianco Marco	0512102462

**Revision History**

Data	Versione	Descrizione	Autore
7/12/2015	1.0	Scelte progettuali dell'Object Design, Linee guida della documentazione	Gaglione Rino
8/12/2015	1.1	Design Pattern	Gaglione Rino
10/12/2015	1.2	Pacchetti	Gaglione Rino
12-18/12/2015	1.3	Interfaccia delle Classi	Gaglione Rino

# Indice

## **1.Introduzione**

- 1.1 Scelte progettuali dell'Object Design
- 1.2 Linee guida della documentazione delle interfacce
- 1.3 Definizioni, acronimi, ed abbreviazioni
- 1.4 Riferimenti
- 1.5 Design Pattern

## **2.Pacchetti**

## **3.Intefacce delle classi**

# **1.Introduzione**

## **1.1 Scelte progettuali dell'Object Design**

### **Trade-off**

#### **Comprensibilità vs costi**

Si preferisce aggiungere costi per la documentazione al fine di rendere il codice comprensibile anche alle persone non coinvolte nel progetto o le persone coinvolte che non hanno lavorato a quella parte in particolare. Commenti diffusi nel codice facilitano la comprensione, di conseguenza migliorare la comprensibilità agevola il mantenimento e anche il processo di modifica.

#### **Interfaccia vs Easy-use**

Il sistema Negozio-Online è molto semplice e di facile utilizzo poiché ha un'interfaccia chiara e intuitiva.

## **1.2 Linee guida per la documentazione dell'interfaccia**

La convenzione per quanto riguarda i nomi dei file, delle operazioni è quella di avere nomi evocativi, ma soprattutto in lingua italiana; per quanto riguarda i nomi delle variabili, è la nota Camel Notation.

#### **Organizzazione dei file**

Ogni file deve essere:

- Sviluppato e diviso in base alla categoria di appartenenza, ovvero deve essere correlato ad un'unica funzionalità che persegue. Ogni pagina del NegoziOnline(login, elenco dei prodotti ,ricerca , modifica Account,registrazione etc.) deve essere implementata in file separati;
- Diviso in più file, se raggiunge una lunghezza tale da divenire difficile da leggere e comprendere;

- La convenzione per quanto riguarda i nomi dei file, delle operazioni e delle variabili è quella di avere nomi evocativi, ma soprattutto in lingua italiana.

### **Spostamento di linee**

Quando un'espressione supera la lunghezza della linea, occorre spezzarla secondo i seguenti principi generali:

- Interrompere la linea dopo una virgola;
- Interrompere la linea prima di un operatore;
- Preferire interruzioni di alto livello rispetto ad interruzioni di basso livello (interrompere laddove non si interrompe un discorso logico, discorso valido soprattutto per le formule es.  $(3+4) * 2$  interrompere prima della moltiplicazione senza spezzare gli operandi in parentesi);
- Allineare la nuova linea con l'inizio dell'espressione nella linea precedente;
- Se le regole precedenti rendono il codice più confuso o il codice è troppo spostato verso il margine destro, utilizzare solo otto spazi di indentazione.

### **Indentazione**

L'indentazione deve essere effettuata con un TAB e qualunque sia il linguaggio usato per la produzione di codice, ogni istruzione deve essere opportunamente indentata.

*Es.*

```
<html>
```

```
<head>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```

Deve essere sostituita da:

```
<html>

    <head>

    </head>

    <body>

    </body>

</html>
```

Questa pratica deve essere usata soprattutto per le istruzioni FOR, IF.

È buona pratica scendere di livello.

### **Inizializzazione**

Inizializzare le variabili locali nel punto in cui sono state dichiarate a meno che il suo valore iniziale non dipenda da un calcolo che occorre eseguire prima.

### **Posizione**

Mettere le dichiarazioni all'inizio dei blocchi. Non aspettare di dichiarare le variabili al loro primo uso: può confondere il programmatore inesperto e impedire la portabilità del codice dentro lo scope. L'unica eccezione a questa regola sono gli indici dei cicli for che in Java possono essere dichiarati nell'istruzione stessa. Evitare dichiarazioni locali che nascondono dichiarazioni a più alto livello. Ad esempio, non dichiarare una variabile con lo stesso nome in un blocco interno.

### **Parentesi**

A prescindere dalle istruzioni che seguono un IF, è necessario, laddove ci fosse anche una sola istruzione, riportare il blocco di istruzioni tra parentesi graffe.

Ogni tag di apertura deve essere necessariamente seguito dall'apposito tag di chiusura (eccetto i tag self-closing).

Una convenzione importante, per quanto riguarda l'inserimento di numeri o di valori costanti, è quella di non usare una codifica fissa (hard coding) che è fortemente sconsigliata ma di associare sempre il valore ad una variabile o semplicemente definire una macro che può essere richiamata da eventi ed essere parametrizzata. In questo

modo si facilita la modifica, sostituendo solo il valore della variabile o macro, in un unico posto.

### **1.3 Definizioni, acronimi, ed abbreviazioni**

#### **Acronimi**

*ODD*: Object Design Document

*DBMS*: Database Management System

#### **Definizioni**

HTML: Linguaggio di mark-up per pagine web.

CSS: Linguaggio usato per definire la formattazione di pagine web.

JavaScript: Linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da *eventi* innescati a loro volta in vari modi dall'utente sulla pagina web in uso.

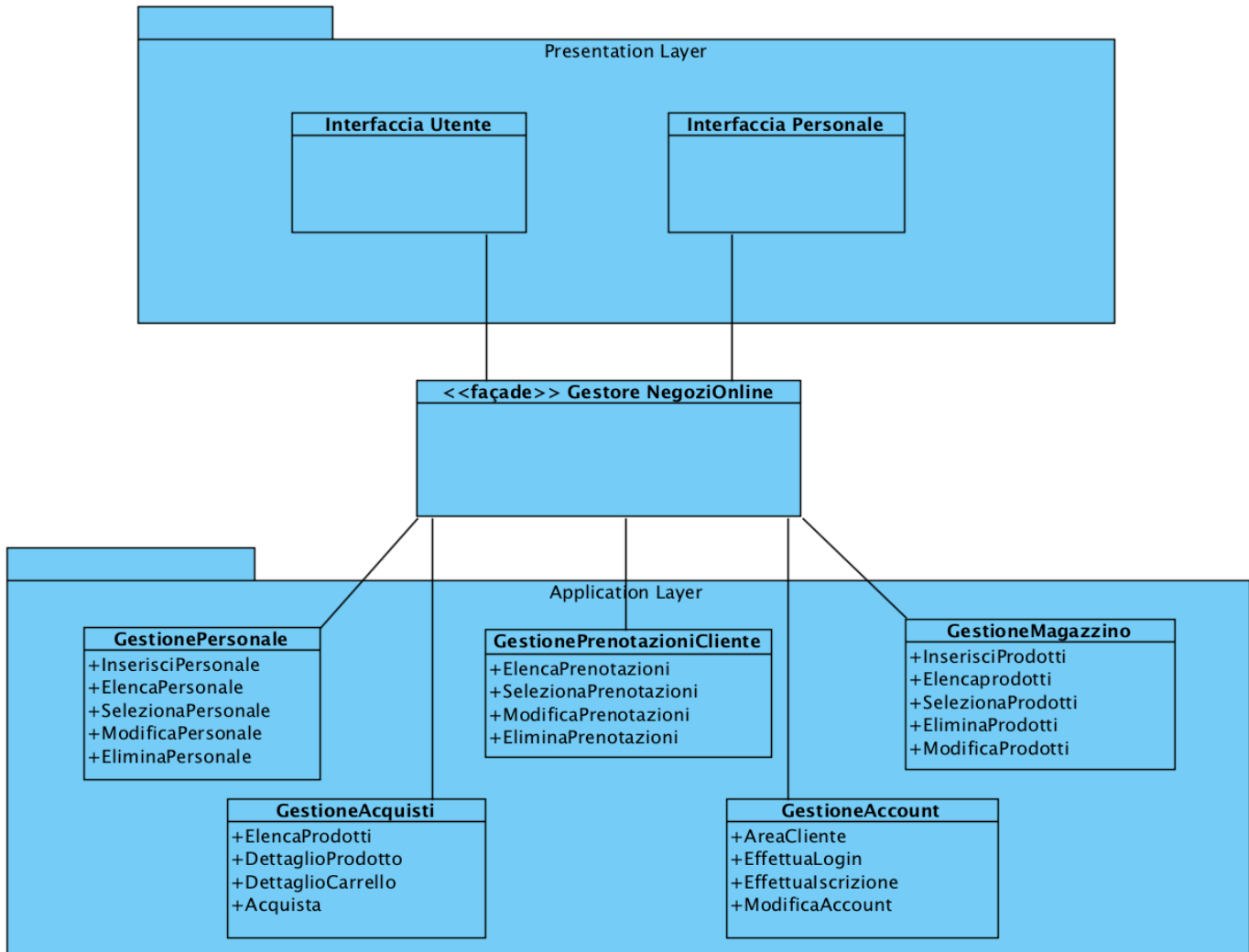
Hard Coding: Codifica fissa.

### **1.4 Riferimenti**

Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (3rd edition), Prentice- Hall, 2003.

## 1.5 Design Pattern

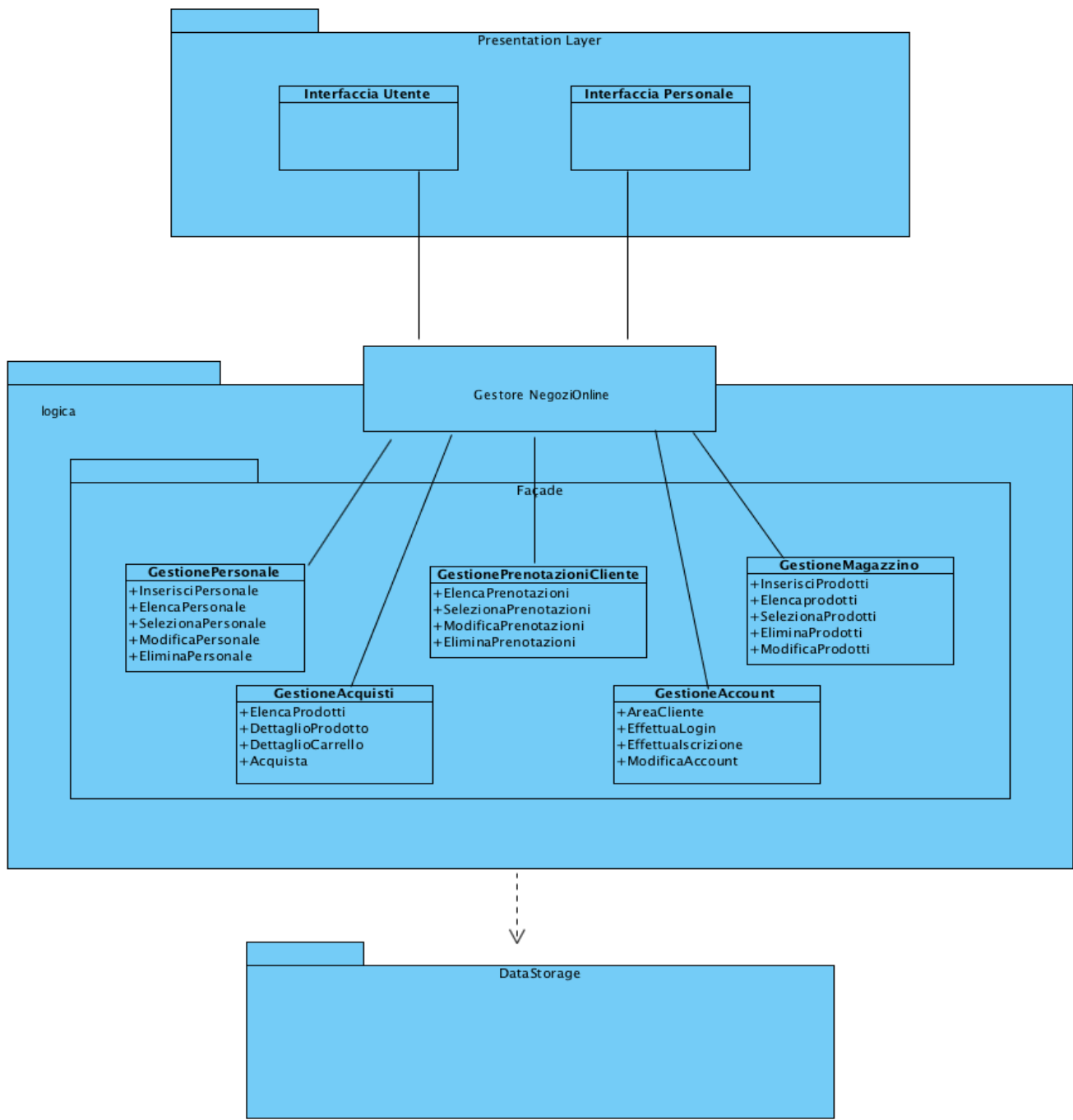
### Facade Pattern



Il sistema Negozio-Online fa uso del Façade Pattern per definire un'unica interfaccia a livello di logica che permette all'utente di interagire, attraverso l'interfaccia, con le funzionalità del sistema vedendole come un unico sistema.



## 2.Pacchetti



### 3. Interfaccia delle classi

Nome classe	<b>GestionePersonale</b>
Descrizione	Questa classe permette l'inserimento, la cancellazione e la modifica del personale
Pre-condizioni	<p><b>context</b> GestionePersonale:: inserisciPersonale(Nome,Cognome,dataNascita,ruolo,username,password); <b>pre:</b> Username!=null &amp;&amp; Password!=null &amp;&amp; dataNascita !=null</p> <p><b>context</b> GestionePersonale: modificaPersonale(Nome,Cognome,dataNascita,ruolo,user); <b>pre:</b> Nome !=null &amp;&amp; Cognome !=null &amp;&amp; dataNascita !=null &amp;&amp; ruolo !=null &amp;&amp; user &amp;&amp; !=null</p> <p><b>context</b> GestionePersonale:eliminaPersonale(IDpersonale); <b>pre:</b> IDpersonale=Id del personale selezionato</p> <p><b>context</b> GestionePersonale:ricercaPersonale(ruolo); <b>pre:</b> ruolo=0    ruolo=1    ruolo=2    ruolo=3</p> <p><b>context</b> GestionePersonale:selezionaPersonale(id); <b>pre:</b> id=idPersonale selezionato</p>
Post-condizioni	<p><b>context</b> GestionePersonale:: inserisciPersonale(Nome,Cognome,dataNascita,ruolo,username,password); <b>post:</b> personale inserito</p> <p><b>context</b> GestionePersonale: modificaPersonale(Nome,Cognome,dataNascita,ruolo,user); <b>post:</b> (Nome =newNome o Cognome =newCognome o dataNascita =newData )    ruolo =newRuolo &amp;&amp; user=newUser</p> <p><b>context</b> GestionePersonale:eliminaPersonale(IDpersonale); <b>post:</b> personale eliminato</p> <p><b>context</b> GestionePersonale:ricercaPersonale(ruolo); <b>post:</b> elenco personale per quel ruolo</p> <p><b>context</b> GestionePersonale:selezionaPersonale(id); <b>post:</b> tabella dettagli del personale con quel id</p>

--	--

Nome classe	<b>GestioneMagazzino</b>
Descrizione	Permette l'aggiunta di nuovi prodotti, la visualizzazione di tutti i prodotti presenti ,la modifica e la cancellazione dei dati relativi ai prodotti.
Pre-condizioni	<p><b>context</b> GestioneMagazzino: inserisciProdotti(IDprodotto,IDs,Nome,Descrizione,Scheda_Tecnica,Prezzo,Immagine); <b>pre:</b> IDprodotto!=null &amp;&amp; IDs !=null &amp;&amp;Nome!=null &amp;&amp;Descrizione!=null &amp;&amp;Scheda_Tecnica!=null &amp;&amp; Prezzo!=null &amp;&amp; Immagine!=null</p> <p><b>context</b> GestioneMagazzino: eliminaProdotti(IDprodotto); <b>pre:</b> IDprodotto=id</p> <p><b>context</b> GestioneMagazzino: modificaProdotto(Idprodotto,Idsottocategoria,nome,descrizione,schedaTecnica,prezzo,giacenza,immagine); <b>pre:</b> Idprodotto!=null &amp;&amp; Idsottocategoria!=null &amp;&amp; nome!=null &amp;&amp; schedaTecnica!=null &amp;&amp; prezzo!=null &amp;&amp; giacenza!=null &amp;&amp; immagine !=null</p> <p><b>context</b> GestioneMagazzino: ricercaProdotti(valore); <b>pre:</b> valore=0(IdArticolo)    valore=1(NomeArticolo)    valore=2(Categoria)    valore=3(Prenotazioni)    valore=4(Giacenza)</p> <p><b>context</b> GestioneMagazzino: selezionaProdotti(idprodotto); <b>pre:</b> idProdotto=id del prodotto selezionato</p>
Post-condizioni	<p><b>context</b> GestioneMagazzino: inserisciProdotti(); <b>post:</b> prodotto inserito</p> <p><b>context</b> GestioneMagazzino: eliminaProdotti(IDprodotto); <b>post:</b> IDprodotto=null</p> <p><b>context</b> GestioneMagazzino: modificaProdotto(Idprodotto,Idsottocategoria,nome,descrizione,schedaTecnica,prezzo,giacenza,immagine); <b>post:</b> Idprodotto=Id &amp;&amp;(Idsottocategoria=newIds,    nome=newNome,    descrizione=newDescrizione    schedaTecnica=newSchedaTecnica    prezzo=newPrezzo,    giacenza=newGiacenza   immagine=newImmagin)</p> <p><b>context</b> GestioneMagazzino: ricercaProdotti(valore); <b>post:</b> elenco dei prodotti per il valore selezionato di ricerca</p> <p><b>context</b> GestioneMagazzino: selezionaProdotti(idprodotto); <b>post:</b> dettagli relativi al prodotto con id=Idprodotto</p>

Nome classe	<b>GestionePrenotazioniCliente</b>
Descrizione	Permette di consultare le prenotazioni effettuate dai Clienti, di apportare modifiche e di cancellarle.
Pre-condizioni	<p><b>context</b> GestionePrenotazioniCliente : ricercaPrenotazioni(valore); <b>pre:</b> valore=0(Data)    valore=1(In attesa)    valore=2(Evase)</p> <p><b>context</b> GestionePrenotazioniCliente :selezionaPrenotazione(IDprenotazione);<b>pre:</b> IDprenotazione=id della prenotazione selezionata</p> <p><b>context</b> GestionePrenotazioniCliente :modificaStato(IDPrenotazione,Stato);<b>pre:</b> IDprenotazione=id prenotazione che è stata visualizzata &amp;&amp;(Stato=0    Stato=1)</p> <p><b>context</b> GestionePrenotazioniCliente :eliminaPrenotazione(IDprenotazione);<b>pre:</b> IDprenotazione=id prenotazione visualizzata</p>
Post-condizioni	<p><b>context</b> GestionePrenotazioniCliente : ricercaPrenotazioni(valore); <b>post:</b> elenco prenotazioni con id=valore.</p> <p><b>context</b> GestionePrenotazioniCliente :selezionaPrenotazione(IDprenotazione);<b>post:</b> dettagli relativi alla prenotazione con id=IDprenotazione</p> <p><b>context</b> GestionePrenotazioniCliente :modificaStato(IDPrenotazione,Stato);<b>post:</b> Stato=newStato</p> <p><b>context</b> GestionePrenotazioniCliente :eliminaPrenotazione(IDprenotazione);<b>post:</b> IDprenotazione=null</p>

Nome classe	<b>GestioneAcquisti</b>
Descrizione	Permette la consultazione dei prodotti disponibili, l'inserimento dei prodotti al carrello ed eventuali modifiche e l'acquisto dei prodotti .
Pre-condizioni	<b>Context</b> GestioneAcquisti: elencaProdotti(IDprodotto); <b>pre:</b> IDprodotto=ID sottocategoria selezionata

	<p><b>Context</b> GestioneAcquisti: dettaglioProdotto(IDprodotto);<b>pre:</b> IDprodotto=ID prodotto selezionato</p> <p><b>Context</b> GestioneAcquisti: Acquista(IDcliente,IDprodotto, quantita);<b>pre:</b> quantita!=0 &amp;&amp; IDcliente!=null &amp;&amp; IDprodotto=id prodotti carrello</p>
Post-condizioni	<p><b>Context</b> GestioneAcquisti: elencaProdotti(IDprodotto);<b>post:</b> elenco dei prodotti per id=IDprodotti selezionato di ricerca</p> <p><b>Context</b> GestioneAcquisti: dettaglioProdotto(IDprodotto);<b>post:</b> dettagli relativi al prodotto con id=IDprodotto</p>

Nome classe	<b>GestioneAccount</b>
Descrizione	permette di visualizzare ed apportare modifiche al proprio account, di iscrizione ed accesso al sistema, e di chiedere eventualmente assistenza.
Pre-condizioni	<p><b>Context</b> GestioneAccount:login(); <b>pre:</b> Username!=null &amp;&amp; password !=null</p> <p><b>Context</b> GestioneAccount:modificaAccount(Nome,Cognome,DataNascita,LuogoResidenza,Provincia,Cap,Via,Email,Username); <b>pre:</b> Nome!=null &amp;&amp; Cognome!=null &amp;&amp; DataNascita!=null &amp;&amp; LuogoResidenza!=null &amp;&amp; Provincia !=null &amp;&amp; Cap!=null &amp;&amp; Via!=null &amp;&amp; Email!=null &amp;&amp; Username=user</p> <p><b>Context</b> GestioneAccount:dettagliAccount(IDcliente);</p> <p><b>Context</b> GestioneAccount:iscrizione(Nome,Cognome,Username&gt;Password,DataNascita,LuogoResidenza,Provincia,Cap,Via,Email); <b>pre:</b> :Nome!=null &amp;&amp; Cognome!=null &amp;&amp; Username!=null &amp;&amp; Password!=null &amp;&amp; DataNascita!=null &amp;&amp; LuogoResidenza!=null &amp;&amp; Provincia !=null &amp;&amp; Cap!=null &amp;&amp; Via!=null &amp;&amp; Email!=null</p>
Post-condizioni	<p><b>Context</b> GestioneAccount: modificaAccount(Nome,Cognome,DataNascita,LuogoResidenza,Provincia,Cap,Via,Email); <b>post:</b> Nome=newNome    Cognome=newCognome    LuogoResidenza=newLuogo    Provincia=newProvincia    Cap=newCap    Via=newVia    Email=newEmail</p>