

***CORSO DI INGEGNERIA DEL SOFTWARE***  
***PROF. GIANCARLO NOTA***  
***A.A. 2006/2007***



**PROGETTO:**            ***G.P.C. (Gestione Polo Conciario)***

**ESEGUITO DA:**

<b>Davino Cristiano:</b>	<b>0510200572</b>
<b>Ferro Amedeo:</b>	<b>0510200578</b>
<b>Mercogliano Umberto:</b>	<b>0510200638</b>
<b>Ferri Vincenzo:</b>	<b>0510200659</b>

**DOCUMENTO:**            **Object Design Document( ODD )**

# PROGETTO : GPC

## DOCUMENTO : DOCUMENTO DI OBJECT DESIGN DOCUMENT

**SCRITTO DA : Davino Cristiano**

**DATA: 27/06/2007**

**Approvato dal team.**

## STRUMENTI C.A.S.E. DI SUPPORTO

STRUMENTO C.A.S.E.	PRODUTTORE	UTILIZZATO PER
WORD 2003	MICROSOFT	STESURA DOCUMENTO
WINPROJECT PRO 2003	MICROSOFT	SCHEDULING PROGETTO
POISEDON FOR UML 5.0	GENTLEWARE	GRAFICI UML
IBM RATIONAL MODELER	IBM	GRAFICI UML

## **1. Riuso**

I componenti identificati durante il system design sono utilizzati per aiutare la realizzazione di ciascun sottosistema. Le librerie di classi e componenti aggiuntivi sono selezionati per le strutture dati e servizi di base. Pattern di design sono selezionati per la risoluzione di problemi comuni e per la protezione di classi specifiche da futuri cambiamenti. Spesso i Patterns e i componenti di design necessitano di essere adattati prima del loro utilizzo. Ciò è fatto con l'impacchettamento degli oggetti attorno ad essi o ridefinendoli utilizzando l'ereditarietà. Durante tutte queste attività, noi sviluppatori abbiamo affrontato tutti i lavori che sono stati rilevati nel system design.

## **2. Specificazione delle interfaccia**

Durante quest'attività i servizi di sottosistema identificati durante il system design sono specificati in termini di interfacce di classe incluse operazioni, argomenti, tipi di firme ed eccezioni. Operazioni aggiuntive e gli oggetti necessitano di trasferire dati tra sottosistemi che sono stati anch'essi identificati. Il risultato del servizio della specificazione è una completa specificazione dell'interfaccia per ciascun sottosistema. La specificazione del servizio del sottosistema è spesso chiamata sottosistema API (Interfaccia di Programmazione di Applicazione).

## **3. Ottimizzazione**

Le attività di ottimizzazione indirizzano i requisiti delle prestazioni di un modello di sistema. Queste includono il cambiamento degli algoritmi per rispondere alla velocità o ai requisiti di memoria, riducendo la cardinalità in associazioni più veloci delle query aggiungendo le associazioni ridondanti per l'efficienza, riallineare l'ordine dell'esecuzione, aggiungendo attributi derivati per provare il tempo d'accesso agli oggetti, e aprendo l'architettura, che l'aggiunta gli accessi ai livelli più bassi per adempiere ai requisiti di prestazioni.

## **4. Linee Guida per la Documentazione delle Interfacce**

Gli sviluppatori dovranno seguire alcune linee guida per la scrittura del codice:

- I nomi dei packages dovranno essere in maiuscolo.
- I nomi delle classi dovranno incominciare tutti con lettera maiuscola e continuare con lettere minuscole ( si ammettono eccezioni di poche lettere, esempio: PaginaAutenticazione() );
- I metodi per l'accesso e la modifica delle variabili dovranno avere rispettivamente il suffisso get (accesso) e set (modifica, inserimento, cancellazione).
- I commenti alle classi, ai metodi e alle variabili di classe dovranno seguire lo standard JavaDoc, quindi iniziare con `/**` e terminare con `*/`.

In particolare useremo i seguenti tag:

`@author` *nome dell'autore*

`@version` *numero di versione della classe*

`<p>Title:` *nome della classe* `</p>`

`<p>Description:` *breve descrizione della classe* `</p>`

`<p>Copyright:` *eventuale copyright* `</p>`

`<p>Company:` *nome della società* `</p>`

## **5. Definizioni Acronimi e Abbreviazioni**

**Packaging (Requisito):** E' un requisito funzionale, rappresenta una particolare restrizione sulla consegna attuale del sistema.

**Requisiti Funzionali:** Un'area di funzionalità che il sistema deve supportare; descrivono le interazioni tra un attore e il sistema a prescindere dalla realizzazione vera e propria del sistema stesso. Ad essi correlati vi è il Modello funzionale.

**Modello Funzionale:** Descrive la funzionalità del sistema dal punto di vista dell'utente.

**Requisiti non Funzionali:** Una costrizione visibile all'utente sul sistema. Descrivono gli aspetti del sistema direttamente visibili dall'utente che non sono associati con la funzionalità del sistema stesso.

**Attore:** Un'entità che interagisce con il sistema eseguendo specifiche operazioni.

**Caso d'uso:** Una sequenza generale di interazioni tra uno o più attori e il sistema.

**Scenario:** Illustra una concreta sequenza di interazioni tra uno o più attori e il sistema; fornisce una descrizione particolareggiata di un determinato Caso d'Uso; descrive l'istante temporale, lo "status" di un particolare Caso d'Uso.

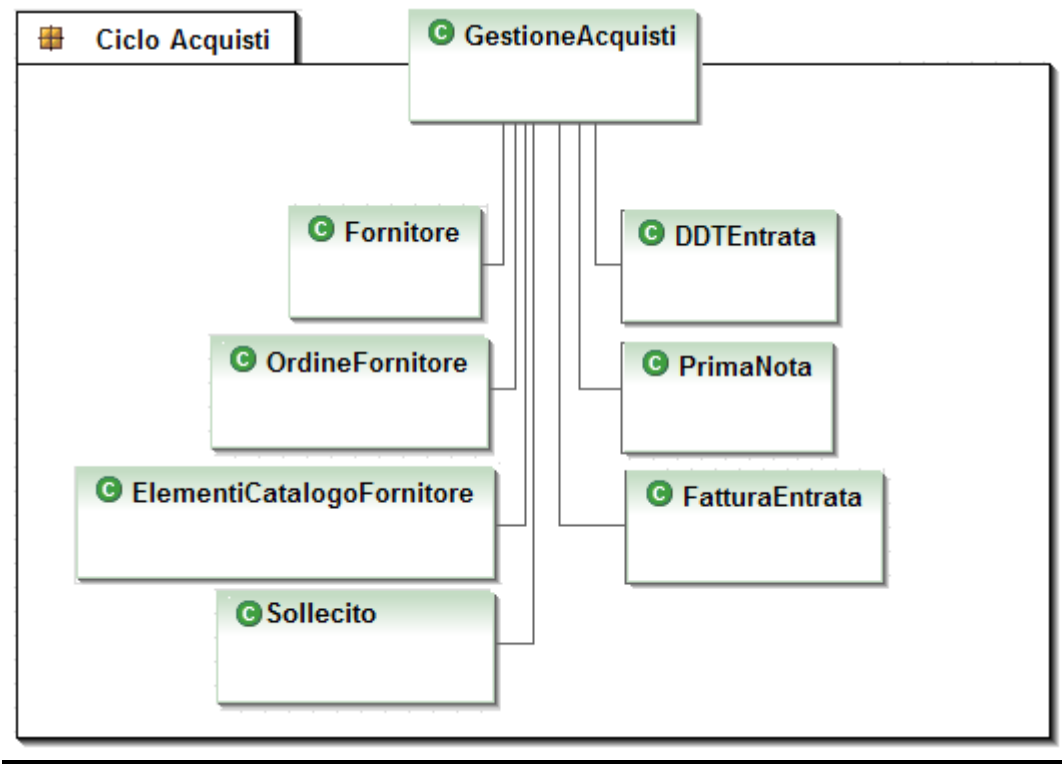
**Magazziniere:** Fornisce l'interfaccia grafica tra il sistema e il magazziniere per iniziare tutti i casi d'uso.

**Gestore:** Fornisce l'interfaccia grafica tra il sistema e il Gestore per iniziare tutti i casi d'uso.

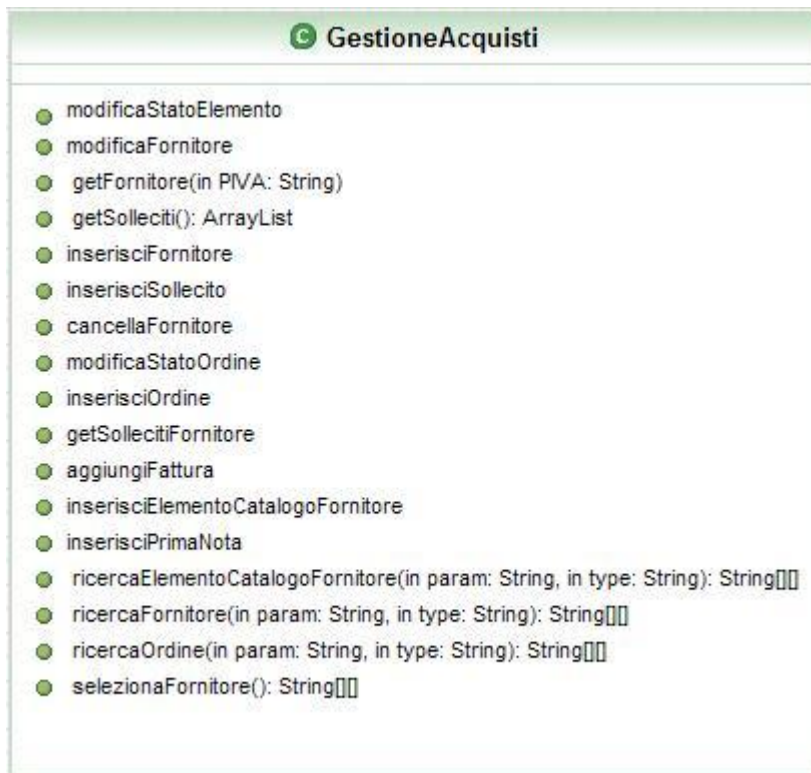
**Storage:** Si occupa della memorizzazione dei dati persistenti. Contiene i dati dei clienti, i dati dei fornitori, gli ordini in corso ed evasi, gli elementi del catalogo azienda e del catalogo fornitore, la merce (pelli) presenti all'interno del magazzino.

**6. DIAGRAMMA DELLE CLASSI**

**6.1 CICLO ACQUISTI**



## Classe GestioneAcquisti



## Elementi Catalogo Fornitore



## Classe Fornitore

### Fornitore

- ▣ Cap: int
- ▣ Categoria: String
- ▣ Cellulare: String
- ▣ Fiducia: String
- ▣ Indirizzo: String
- ▣ Località: String
- ▣ PagamentoPredefinito: String
- ▣ PartitaIVA: String
- ▣ Provincia: String
- ▣ RagioneSociale: String
- ▣ Rappresentante: String
- ▣ ScontoPredefinito: double
- ▣ Telefono: String

- getCap(): int
- getCategory(): String
- getCellulare(): String
- getFiducia(): String
- getIndirizzo(): String
- getLocalità(): String
- getPagamentoPredefinito(): String
- getPartitaIVA(): String
- getProvincia(): String
- getRagioneSociale(): String
- getRappresentante(): String
- getScontoPredefinito(): double
- getTelefono(): String
- setCap(in cap: int)
- setCategoria(in categoria: String)
- setCellulare(in cellulare: String)
- setFiducia(in fiducia: String)
- setIndirizzo(in indirizzo: String)
- setLocalità(in località: String)
- setPagamentoPredefinito(in pagamentoPredefinito: String)
- setPartitaIVA(in partitaIVA: String)
- setProvincia(in provincia: String)
- setRagioneSociale(in ragioneSociale: String)
- setRappresentante(in rappresentante: String)
- setScontoPredefinito(in scontoPredefinito: double)
- setTelefono(in telefono: String)



## Classe OrdineFornitore

C OrdineFornitore
<ul style="list-style-type: none"><li>Elementi: ElementiCatalogoFornitore</li><li>idFornitore: int</li><li>idOrdine: int</li><li>imponibile: double</li><li>DataOrdine: String</li><li>DataArrivo: String</li></ul>
<ul style="list-style-type: none"><li>getElementi(): ArrayList</li><li>getIdFornitore(): int</li><li>getIdOrdine(): int</li><li>getImponibile(): double</li><li>getDataArrivo(): String</li><li>setIdFornitore(in idFornitore: int)</li><li>setIdOrdine(in idOrdine: int)</li><li>setImponibile(in imponibile: double)</li><li>getDataOrdine(): String</li><li>setDataOrdine(in dataOrdine: String)</li><li>setDataArrivo(in dataArrivo: String)</li></ul>

## Classe DDTEntrata

C DDTEntrata
<ul style="list-style-type: none"><li>Data: String</li><li>id: int</li><li>idFattEntrata: int</li></ul>
<ul style="list-style-type: none"><li>getData(): String</li><li>getId(): int</li><li>getIdFattEntrata(): int</li><li>setData(in data: String)</li><li>setId(in id: int)</li><li>setIdFattEntrata(in idFattEntrata: int)</li></ul>

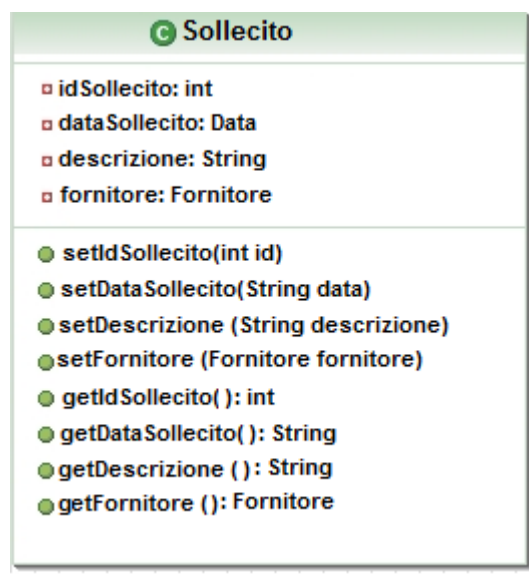
## Classe PrimaNota

C PrimaNota	
▣ avere: double	
▣ dare: double	
▣ Descrizione: String	
▣ idPrimaNota: int	
● getAvere(): double	
● getDare(): double	
● getDescrizione(): String	
● getIdPrimaNota(): int	
● setAvere(in avere: double)	
● setDare(in dare: double)	
● setDescription(in descrizione: String)	
● setIdPrimaNota(in idPrimaNota: int)	

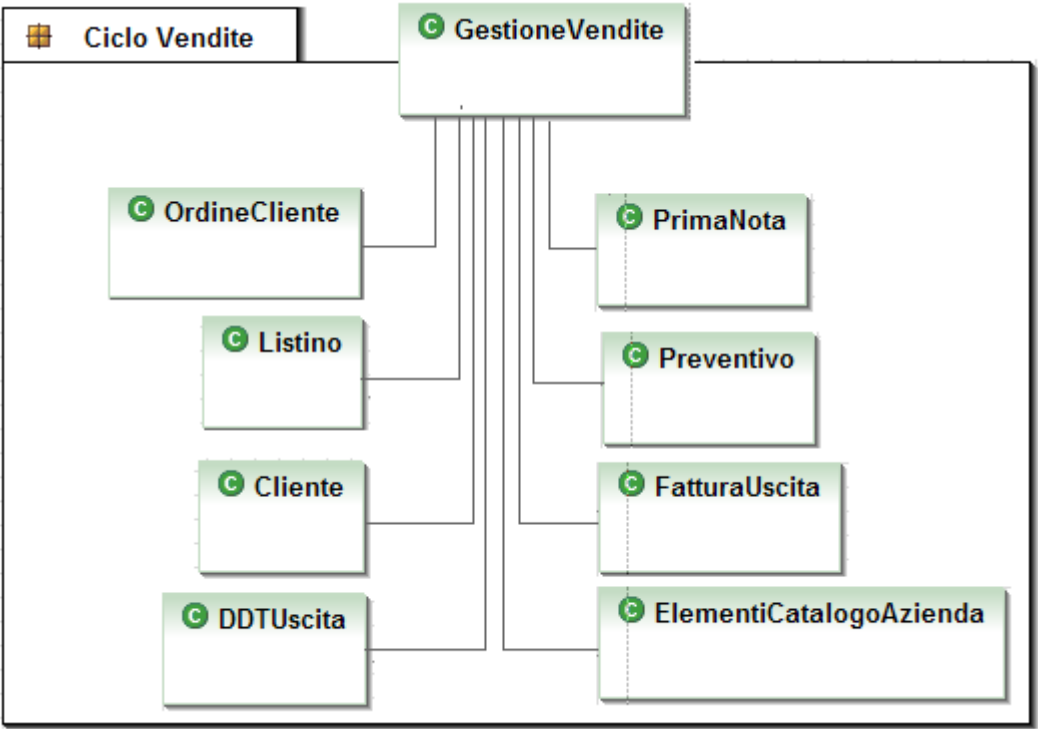
## Classe FatturaEntrata

C FatturaEntrata	
▣ data: String	
▣ Iva: double	
▣ Mittente: Fornitore	
▣ Totale: double	
▣ DDT_In: DDTEntrata	
▣ numFattura: int	
● getDDT_In(): ArrayList	
● getData(): String	
● getIva(): double	
●.getMittente(): Fornitore	
● getTotale(): double	
● setData(in data: String)	
● setIva(in iva: double)	
● setMittente(in mittente: Fornitore)	
● setTotale(in totale: double)	
● setNumFattura(in numFattura: int)	
● getNumFattura(): int	

## Classe Sollecito



**6.2 CICLO VENDITE**



## Classe GestioneVendite

GestioneVendite
<ul style="list-style-type: none"><li>● aggiungiElementoCatalogoAzienda</li><li>● aggiungiFattura</li><li>● aggiungiListino</li><li>● cancellaCliente</li><li>● cancellaPreventivo</li><li>● eliminaElementoCatalogoAzienda</li><li>● eliminaListino</li><li>● getCliente(in CF: String)</li><li>● getPreventivo</li><li>● inserisciCliente</li><li>● inserisciOrdine</li><li>● inserisciPreventivo</li><li>● inserisciPrimaNota</li><li>● modificaCliente</li><li>● modificaStatoOrdine</li><li>● ricercaCliente(in param: String, in type: String): String[]</li><li>● ricercaElementoCatalogoAzienda(in param: String, in type: String): String[]</li><li>● ricercaOrdine(in param: String, in type: String): String[]</li><li>● selezionaCliente(): String[]</li></ul>

## Classe Cliente

### Cliente

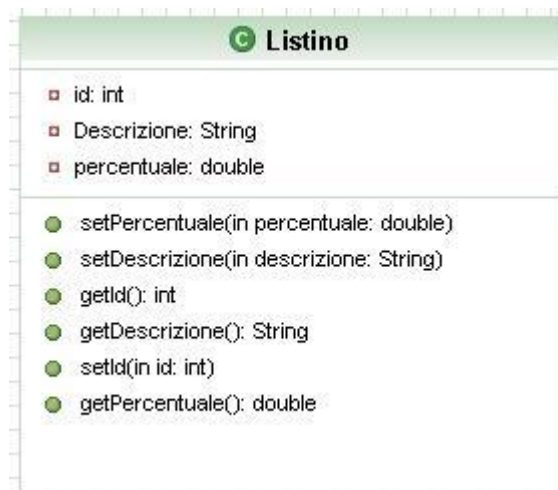
- Cap: int
- Categoria: String
- Cellulare: String
- Codice: int
- CodiceFiscale: String
- Cognome: String
- Fiducia: double
- Indirizzo: String
- Località: String
- Nome: String
- PagamentoPredefinito: String
- Provincia: String
- RagioneSociale: String
- Telefono: String

- getCap(): int
- getCategory(): String
- getCellulare(): String
- getCodice(): int
- getCodiceFiscale(): String
- getCognome(): String
- getFiducia(): double
- getIndirizzo(): String
- getLocalità(): String
- getNome(): String
- getPagamentoPredefinito(): String
- getProvincia(): String
- getRagioneSociale(): String
- getTelefono(): String
- setCap(in cap: int)
- setCategoria(in categoria: String)
- setCellulare(in cellulare: String)
- setCodice(in codice: int)
- setCodiceFiscale(in codiceFiscale: String)
- setCognome(in cognome: String)
- setFiducia(in fiducia: double)
- setIndirizzo(in indirizzo: String)
- setLocalità(in località: String)
- setNome(in nome: String)
- setPagamentoPredefinito(in pagamentoPredefinito: String)
- setProvincia(in provincia: String)
- setRagioneSociale(in ragioneSociale: String)
- setTelefono(in telefono: String)

## Classe OrdineCliente



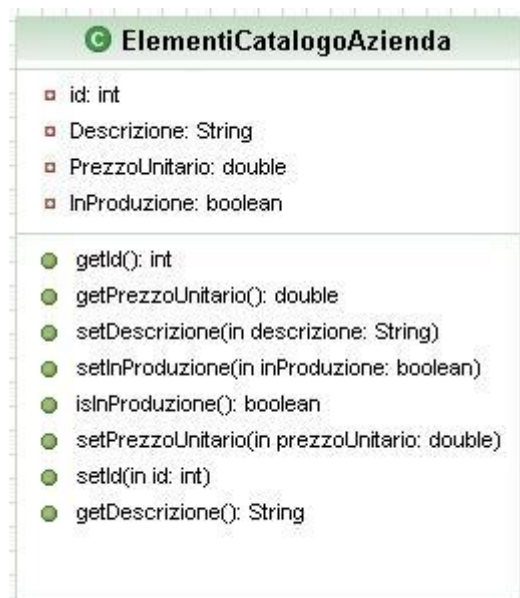
## Classe Listino



## Classe DDTUscita



## Classe ElementiCatalogoAzienda





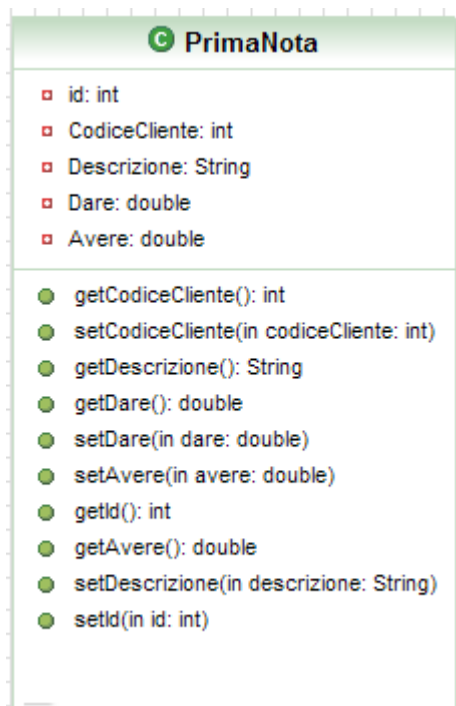
## Classe FatturaUscita



## Classe Preventivo



## Classe PrimaNota



# Class GestioneAcquisti

java.lang.Object

└─ **GestioneAcquisti**

```
public class GestioneAcquisti
extends java.lang.Object
```

**Author:**  
mbi

Constructor Summary	
<a href="#">GestioneAcquisti</a> ()	

Method Summary	
void	<a href="#">aggiungiFattura</a> (Fattura aFattura, java.util.ArrayList<DDTEntrata> array) Aggiunge una fattura nel DataBase emessa da un determinato fornitore allegando i relativi DDT.
void	<a href="#">cancellaFornitore</a> (Fornitore aFornitore) Cancella un fornitore già presente nel DataBase.
Fornitore	<a href="#">getFornitore</a> (java.lang.String PIVA) Visuallizza le informazioni di un fornitore.
java.util.ArrayList<Sollecito>	<a href="#">getSolleciti</a> () Visuallizza i solleciti presenti nel DataBase.
java.util.ArrayList<Sollecito>	<a href="#">getSollecitiFornitore</a> (Fornitore aFornitore) Visuallizza le informazioni sui solleciti di un fornitore.
void	<a href="#">inserisciElementoCatalogoFornitore</a> (Fornitore aFornitore, ElementiCatalogoFornitore elem) Inserisce un nuovo elemento nel catalogo del relativo fornitore.
void	<a href="#">inserisciFornitore</a> (Fornitore aFornitore) Inserisce un fornitore nel DataBase.
void	<a href="#">inserisciOrdine</a> (Fornitore aFornitore, java.util.ArrayList<ElementiCatalogoFornitore> array) Inserisce un ordine nel DataBase richiesto ad un determinato fornitore.
void	<a href="#">inserisciPrimaNota</a> (Fornitore aFornitore, PrimaNota aPrimaNota) Inserisce un'operazione di prima nota nel DataBase relato alla scheda contabile del fornitore.
void	<a href="#">inserisciSollecito</a> (Fornitore aFornitore, java.util.ArrayList<Sollecito> array)

	Inserisce un sollecito ad un determinato fornitore.
void	<a href="#"><u>modificaFornitore</u></a> (Fornitore aFornitore) Modifica un fornitore già presente nel DataBase.
void	<a href="#"><u>modificaStatoElemento</u></a> (ElementiCatalogoFornitore elem) Modifica lo stato di un elemento presente nel catalogo del relativo fornitore.
void	<a href="#"><u>modificaStatoOrdine</u></a> (Ordine aOrdine) Modifica lo stato di un ordine presente nel DataBase ( API ).
java.lang.String[][]	<a href="#"><u>ricercaElementoCatalogoFornitore</u></a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase un determinato prodotto in base a type e param di un fornitore.
java.lang.String[][]	<a href="#"><u>ricercaFornitore</u></a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase dei fornitori in base a type e param.
java.lang.String[][]	<a href="#"><u>ricercaOrdine</u></a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase degli ordini in base a type e param.
java.lang.String[][]	<a href="#"><u>selezionaFornitore</u></a> () Interroga il DataBase su i fornitori memorizzati.

#### Methods inherited from class java.lang.Object

`equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### GestioneAcquisti

```
public GestioneAcquisti()
```

## Method Detail

### inserisciFornitore

```
public void inserisciFornitore(Fornitore aFornitore)
```

Inserisce un fornitore nel DataBase.

#### Parameters:

aFornitore -

---

### modificaFornitore

```
public void modificaFornitore(Fornitore aFornitore)
```

Modifica un fornitore già presente nel DataBase.

#### Parameters:

aFornitore -

---

## **cancellaFornitore**

```
public void cancellaFornitore(Fornitore aFornitore)
    Cancella un fornitore già presente nel DataBase.
Parameters:
    aFornitore -
```

---

## **selezionaFornitore**

```
public java.lang.String[][] selezionaFornitore()
    Interroga il DataBase su i fornitori memorizzati.
Returns:
    String[][] contenente informazioni su tutti i fornitori
```

---

## **getFornitore**

```
public Fornitore getFornitore(java.lang.String PIVA)
    Visualizza le informazioni di un fornitore.
Parameters:
    PIVA -
Returns:
    Un oggetto di tipo Fornitore contenente tutte le informazioni relativo al selezionato
```

---

## **ricercaFornitore**

```
public java.lang.String[][] ricercaFornitore(java.lang.String param,
                                              java.lang.String type)
    Ricerca nel DataBase dei fornitori in base a type e param.
Parameters:
    param -
    type -
Returns:
    String[][] contenente informazioni sui fornitori ricercati.
```

---

## **aggiungiFattura**

```
public void aggiungiFattura(Fattura aFattura,
                           java.util.ArrayList<DDTEntrata> array)
    Aggiunge una fattura nel DataBase emessa da un determinato fornitore allegando i relativi DDT.
Parameters:
    aFattura -
    array -
```

---

## **inserisciOrdine**

```
public void inserisciOrdine(Fornitore aFornitore,
```

```
java.util.ArrayList<ElementiCatalogoFornitore> array)
```

Inserisce un ordine nel DataBase richiesto ad un determinato fornitore.

**Parameters:**

aFornitore -

array -

---

## ricercaOrdine

```
public java.lang.String[][] ricercaOrdine(java.lang.String param,  
                                           java.lang.String type)
```

Ricerca nel DataBase degli ordini in base a type e param.

**Parameters:**

param -

type -

**Returns:**

String[][] contenente informazioni sugli ordini ricercati.

---

## modificaStatoOrdine

```
public void modificaStatoOrdine(Ordine aOrdine)
```

Modifica lo stato di un ordine presente nel DataBase ( API ).

**Parameters:**

aOrdine -

---

## inserisciPrimaNota

```
public void inserisciPrimaNota(Fornitore aFornitore,  
                               PrimaNota aPrimaNota)
```

Inserisce un'operazione di prima nota nel DataBase relativo alla scheda contabile del fornitore.

**Parameters:**

aFornitore -

aPrimaNota -

---

## inserisciElementoCatalogoFornitore

```
public void inserisciElementoCatalogoFornitore(Fornitore aFornitore,  
                                                ElementiCatalogoFornitore elem)
```

Inserisce un nuovo elemento nel catalogo del relativo fornitore.

**Parameters:**

aFornitore -

elem -

---

## modificaStatoElemento

```
public void modificaStatoElemento(ElementiCatalogoFornitore elem)
```

Modifica lo stato di un elemento presente nel catalogo del relativo fornitore.

**Parameters:**

elem -

---

**ricercaElementoCatalogoFornitore**

```
public java.lang.String[][]  
ricercaElementoCatalogoFornitore(java.lang.String param,
```

```
java.lang.String type)
```

Ricerca nel DataBase un determinato prodotto in base a type e param di un fornitore.

**Parameters:**

param -

type -

**Returns:**

String[][] contenente informazioni sul prodotto ricercato.

---

**inserisciSollecito**

```
public void inserisciSollecito(Fornitore aFornitore,  
                               java.util.ArrayList<Sollecito> array)
```

Inserisce un sollecito ad un determinato fornitore.

**Parameters:**

aFornitore -

array -

---

**getSollecitiFornitore**

```
public java.util.ArrayList<Sollecito>  
getSollecitiFornitore(Fornitore aFornitore)
```

Visuallizza le informazioni sui solleciti di un fornitore.

**Parameters:**

aFornitore -

**Returns:**

Un ArrayList di tipo Sollecito contenente tutti i solleciti relativi al selezionato

---

**getSolleciti**

```
public java.util.ArrayList<Sollecito> getSolleciti()
```

Visuallizza i solleciti presenti nel DataBase.

**Returns:**

Un ArrayList di tipo Sollecito contenente tutti i solleciti

---

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class GestioneVendite

java.lang.Object  
└─ **GestioneVendite**

```
public class GestioneVendite
extends java.lang.Object
```

Constructor Summary	
<a href="#">GestioneVendite</a> ()	

Method Summary	
void	<a href="#">aggiungiElementoCatalogoAzienda</a> (OrdineCliente aOrdine) Inserisce un elemento presente nel nostro catalogo azienda all'interno dell'Ordine del Cliente
void	<a href="#">aggiungiFattura</a> (Fattura aFattura, java.util.ArrayList<DDTUscita> array) Aggiunge una fattura nel DataBase emessa ad un determinato Cliente allegando i relativi DDT.
void	<a href="#">aggiungiListino</a> (Cliente aCliente) Aggiunge un Listino al Cliente che passiamo in input
void	<a href="#">cancellaCliente</a> (Cliente aCliente) Cancella un cliente già presente nel DataBase.
void	<a href="#">cancellaPreventivo</a> (Preventivo aPreventivo) Rimuove un Preventivo dal Nostro Database
void	<a href="#">eliminaElementoCatalogoAzienda</a> (ElementiCatalogoAzienda aElementiCatalogoAzienda) Rimuove l'elemento ricevuto come parametro dagli elementi del catalogo dell'azienda
void	<a href="#">eliminaListino</a> (Listino aListino) Rimuove un Listino dal Nostro Database
Cliente	<a href="#">getCliente</a> (java.lang.String CF) Visualizza le informazioni di un cliente.
java.util.ArrayList<Preventivo>	<a href="#">getPreventivo</a> (Cliente aCliente) Ritorna una lista dei preventivi richiesti dal cliente in questione
void	<a href="#">inserisciCliente</a> (Cliente aCliente) Inserisce un cliente nel DataBase.
void	<a href="#">inserisciOrdine</a> (Cliente aCliente, java.util.ArrayList<ElementiCatalogoAzienda> array) Inserisce un ordine nel DataBase richiesto ad un determinato cliente
void	<a href="#">inserisciPreventivo</a> (Cliente cliente, java.util.ArrayList<ElementiCatalogoAzienda> array)



	Inserisce un Preventivo ad un Cliente in base ad una lista della m ordinata
void	<a href="#"><u>inserisciPrimaNota</u></a> (Cliente aCliente, PrimaNota aPrimaNota Inserisce un'operazione di prima nota nel DataBase relato alla sch contabile del cliente.
void	<a href="#"><u>modificaCliente</u></a> (Cliente aCliente) Modifica un cliente già presente nel DataBase.
void	<a href="#"><u>modificaStatoOrdine</u></a> (Ordine aOrdine) Modifica lo stato di un ordine presente nel DataBase ( API ).
java.lang.String[][]	<a href="#"><u>ricercaCliente</u></a> (java.lang.String param, java.lang.String t Ricerca nel DataBase un determinato cliente in base a type e para
java.lang.String[][]	<a href="#"><u>ricercaElementoCatalogoAzienda</u></a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase un determinato prodotto in base a type e pa
java.lang.String[][]	<a href="#"><u>ricercaOrdine</u></a> (java.lang.String param, java.lang.String ty Ricerca nel DataBase un determinato ordine in base a type e para
java.lang.String[][]	<a href="#"><u>selezionaCliente</u></a> () Seleziona dal DataBase un determinato cliente.

#### Methods inherited from class java.lang.Object

equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### GestioneVendite

```
public GestioneVendite()
```

## Method Detail

### inserisciPrimaNota

```
public void inserisciPrimaNota(Cliente aCliente,  
                               PrimaNota aPrimaNota)
```

Inserisce un'operazione di prima nota nel DataBase relato alla scheda contabile del cliente.

#### Parameters:

aCliente -  
aPrimaNota -

### inserisciCliente

```
public void inserisciCliente(Cliente aCliente)
```

Inserisce un cliente nel DataBase.

#### Parameters:

aFornitore -

## modificaCliente

```
public void modificaCliente(Cliente aCliente)
    Modifica un cliente già presente nel DataBase.
Parameters:
    aFornitore -
```

---

## cancellaCliente

```
public void cancellaCliente(Cliente aCliente)
    Cancella un cliente già presente nel DataBase.
Parameters:
    aCliente -
```

---

## selezionaCliente

```
public java.lang.String[][] selezionaCliente()
    Seleziona dal DataBase un determinato cliente.
Returns:
    String[][] contenente informazioni sul cliente scelto
```

---

## getCliente

```
public Cliente getCliente(java.lang.String CF)
    Visualizza le informazioni di un cliente.
Parameters:
    CF -
Returns:
    Un oggetto di tipo Cliente contenente tutte le informazioni relativo al selezionato
```

---

## ricercaCliente

```
public java.lang.String[][] ricercaCliente(java.lang.String param,
                                             java.lang.String type)
    Ricerca nel DataBase un determinato cliente in base a type e param.
Parameters:
    param -
    type -
Returns:
    String[][] contenente informazioni sul cliente ricercato.
```

---

## inserisciOrdine

```
public void inserisciOrdine(Cliente aCliente,
                             java.util.ArrayList<ElementiCatalogoAzienda> array)
    Inserisce un ordine nel DataBase richiesto ad un determinato cliente.
Parameters:
```

aCliente -  
array -

---

## ricercaOrdine

```
public java.lang.String[][] ricercaOrdine(java.lang.String param,  
                                           java.lang.String type)
```

Ricerca nel DataBase un determinato ordine in base a type e param.

**Parameters:**

param -

type -

**Returns:**

String[][] contenente informazioni sull'ordine ricercato.

---

## modificaStatoOrdine

```
public void modificaStatoOrdine(Ordine aOrdine)
```

Modifica lo stato di un ordine presente nel DataBase ( API ).

**Parameters:**

aOrdine -

---

## inserisciPreventivo

```
public void inserisciPreventivo(Cliente cliente,
```

```
java.util.ArrayList<ElementiCatalogoAzienda> array)
```

Inserisce un Preventivo ad un Cliente in base ad una lista della merce ordinata

**Parameters:**

cliente -

array -

---

## cancellaPreventivo

```
public void cancellaPreventivo(Preventivo aPreventivo)
```

Rimuove un Preventivo dal Nostro Database

**Parameters:**

aPreventivo -

---

## getPreventivo

```
public java.util.ArrayList<Preventivo> getPreventivo(Cliente aCLiente)
```

Ritorna una lista dei preventivi richiesti dal cliente in questione

**Parameters:**

aCLiente -

**Returns:**

ArrayList di preventivi fatti al Cliente

---

## aggiungiElementoCatalogoAzienda

```
public void aggiungiElementoCatalogoAzienda(OrdineCliente aOrdine)  
    Inserisce un elemento presente nel nostro catalogo azienda all'interno dell'Ordine del Cliente  
Parameters:  
    aOrdine -
```

---

## eliminaElementoCatalogoAzienda

```
public void eliminaElementoCatalogoAzienda(ElementiCatalogoAzienda aElemento)  
    Rimuove l'elemento ricevuto come parametro dagli elementi del Catalogo dell'azienda  
Parameters:  
    aElemento -
```

---

## aggiungiListino

```
public void aggiungiListino(Cliente aCliente)  
    Aggiunge un Listino al CLiente che passiamo in input  
Parameters:  
    aCliente -
```

---

## eliminaListino

```
public void eliminaListino(Listino aListino)  
    Rimuove un Listino dal Nostro Database  
Parameters:  
    aListino -
```

---

## ricercaElementoCatalogoAzienda

```
public java.lang.String[][]  
ricercaElementoCatalogoAzienda(java.lang.String param,  
  
java.lang.String type)  
    Ricerca nel DataBase un determinato prodotto in base a type e param  
Parameters:  
    param -  
    type -  
Returns:  
    String[][] contenente informazioni sul prodotto ricercato.
```

---

## aggiungiFattura

```
public void aggiungiFattura(Fattura aFattura,  
                             java.util.ArrayList<DDTUscita> array)  
    Aggiunge una fattura nel DataBase emessa ad un determinato Cliente allegando i relativi  
    DDT.
```

**Parameters:**

aFattura -  
array -

[Package](#) [Class](#) [Use](#) [Tree](#) [Deprecated](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

# Class ServiziForniti\_G2

java.lang.Object  
└─ ServiziForniti\_G2

```
public class ServiziForniti_G2
extends java.lang.Object
```

Constructor Summary	
<a href="#">ServiziForniti_G2</a>	()

Method Summary	
Cliente	<a href="#">getCliente</a> (java.lang.String CF) Visuallizza le informazioni di un cliente.
Fornitore	<a href="#">getFornitore</a> (java.lang.String PIVA) Visuallizza le informazioni di un fornitore.
void	<a href="#">modificaStatoOrdineCliente</a> (Ordine aOrdine) Modifica lo stato di un ordine presente nel DataBase ( API ).
void	<a href="#">modificaStatoOrdineFornitore</a> (Ordine aOrdine) Modifica lo stato di un ordine a fornitore presente nel DataBase.
java.lang.String[][]	<a href="#">ricercaOrdineCliente</a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase un determinato ordine in base a type e param.
java.lang.String[][]	<a href="#">ricercaOrdineFornitore</a> (java.lang.String param, java.lang.String type) Ricerca nel DataBase degli ordini ai fornitori in base a type e param.

Methods inherited from class java.lang.Object
<code>equals, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</code>

## Constructor Detail

### ServiziForniti\_G2

```
public ServiziForniti_G2()
```

## Method Detail

## getFornitore

```
public Fornitore getFornitore(java.lang.String PIVA)
```

Visualizza le informazioni di un fornitore.

**Parameters:**

PIVA -

**Returns:**

Un oggetto di tipo Fornitore contenente tutte le informazioni relativo al selezionato

---

## ricercaOrdineFornitore

```
public java.lang.String[][] ricercaOrdineFornitore(java.lang.String param,  
                                                    java.lang.String type)
```

Ricerca nel DataBase degli ordini ai fornitori in base a type e param.

**Parameters:**

param -

type -

**Returns:**

String[][] contenente informazioni sugli ordini ricercati.

---

## modificaStatoOrdineFornitore

```
public void modificaStatoOrdineFornitore(Ordine aOrdine)
```

Modifica lo stato di un ordine a fornitore presente nel DataBase. Limitato a "Ricevuto in Valutazione", "Accettato", "Rifiutato"

**Parameters:**

aOrdine -

---

## getCliente

```
public Cliente getCliente(java.lang.String CF)
```

Visualizza le informazioni di un cliente.

**Parameters:**

CF -

**Returns:**

Un oggetto di tipo Cliente contenente tutte le informazioni relativo al selezionato

---

## ricercaOrdineCliente

```
public java.lang.String[][] ricercaOrdineCliente(java.lang.String param,  
                                                    java.lang.String type)
```

Ricerca nel DataBase un determinato ordine in base a type e param.

**Parameters:**

param -

type -

**Returns:**

String[][] contenente informazioni sull'ordine ricercato.

---

# modificaStatoOrdineCliente

```
public void modificaStatoOrdineCliente(Ordine aOrdine)
    Modifica lo stato di un ordine presente nel DataBase ( API ).
Parameters:
    aOrdine -
```

<a href="#">Package</a>	<a href="#">Class</a>	<a href="#">Use</a>	<a href="#">Tree</a>	<a href="#">Deprecated</a>	<a href="#">Index</a>	<a href="#">Help</a>
<a href="#">PREV CLASS</a>	<a href="#">NEXT CLASS</a>					
SUMMARY: <a href="#">NESTED</a>   <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>				<a href="#">FRAMES</a>   <a href="#">NO FRAMES</a>   <a href="#">All Classes</a>		
				DETAIL: <a href="#">FIELD</a>   <a href="#">CONSTR</a>   <a href="#">METHOD</a>		