Elaborazione Iterazione 2

**Introduzione**

In questa iterazione si è deciso di implementare lo scenario di successo del caso d’uso UC2 **Gestisci Ordine**. Al termine di tale iterazione, Il software sarà in grado di prendere in carico gli ordini dei clienti e di aggiungerli nel sistema.

**Analisi Orientata agli Oggetti**

**Caso d’uso UC2:**

Dopo aver analizzato il caso d’uso si è deciso di aggiungere le seguenti nuove classi concettuali nel modello di dominio:

* **Ordine:** ordine relativo ad un cliente che vuole acquistare delle piante.
* **Spedizione:** rappresenta il tipo di spedizione che il cliente sceglie per ricevere le piante ordinate.
* **Pagamento:** Tipologia di pagamento scelta dal cliente per pagare l’ordine.
* **Archivio:** Archivio dove vengono registrati gli ordini effettuati dai clienti.

Il modello di dominio aggiornato risulta quindi il seguente:

Immagine che contiene testo, mappa

Descrizione generata automaticamente

**Diagrammi di sequenza di sistema**

Di seguito viene riportato il diagramma di sequenza di sistema relativo allo scenario di successo del caso d’uso UC2

Immagine che contiene screenshot

Descrizione generata automaticamente

**Contratti delle operazioni**

Di seguito vengono riportati i contratti relativi alle operazioni principali individuate dai diagrammi di sequenza di sistema.

**Contratto CO1: creaNuovoOrdine**

**Operazione:** creaNuovoOrdine(Cliente *cliente*)

**Riferimenti:** Caso d’uso Gestisci Ordine

**Pre-Condizioni:**

* È presente un’istanza **cliente** di **Cliente**

È presente un’istanza **archivio** di **Archivio**

**Post Condizioni:**

* È stata creata un’istanza **ordine** di **Ordine**
* È stata creata una lista **righeDiOrdine[i]**:**RigaDiOrdine**
* **cliente** è stato associato ad **ordine**

**Contratto CO2: aggiungiPianta**

**Operazione:** aggiungiPianta(Ordine *ordine*, String *tipo*, String *varietà*, int *quantità*, Pianta *pianta* )

**Riferimenti:** Caso d’uso Gestisci ordine

**Pre-Condizioni:**

* è presente un’istanza **o** di **Ordine**
* è presente una **listaRigheDiOrdine[i]:RigaDiOrdine**
* è presente un’istanza **pianta** di **Pianta**

**Post Condizioni:**

* è stata creata un’istanza **rdo** di **RigaDiOrdine**
* L’attributo “quantità” di **rdo** è stato aggiornato con *quantità*
* L’attributo “pianta” di **rdo** è stato aggiornato con *pianta*
* L’attributo “descrizione” di **rdo** è stato aggiornato con la concatenazione di *tipo* e *varietà*
* L’istanza **rdo** è aggiunta a **listaRigheDiOrdine**
* L’attributo quantitàDisponibile di **pianta** è stato aggiornato sottraendo *quantità*

**Contratto C03: Conferma Ordine**

**Operazione:** confermaOrdine(Ordine *ordine*, Cliente *cliente*)

**Riferimenti:** Caso d’uso Gestisci Ordine

**Pre-Condizioni:**

* È presente una **listaOrdiniCliente[i]:Ordine**
* è presente un **archivioOrdini[i]:Ordine**

**Post-Condizioni:**

* L’attributo “totale” relativo all’istanza *ordine* è stato aggiornato
* *ordine* è stato aggiunto a **listaOrdiniCliente**
* *ordine* è stato aggiunto a **archivioOrdini**

**Contratto C04: Seleziona Spedizione**

**Operazione:** selezionaSpedizione(String *tipoSpedizione*)

**Riferimenti:** Caso d’uso Gestisci ordine

**Pre-Condizioni:**

* è presente un’istanza **ordine** di Ordine

**Post Condizioni:**

* è stata creata un’istanza **s** di Spedizione
* l’attributo “tipoSpedzione” di **s** viene aggiornato con *tipoSpedizione*
* l’istanza **s** è stata associata ad **ordine**

**Contratto C05: Seleziona Pagamento**

**Operazione:** selezionaPagamento(String tipoPagamento)

**Riferimenti:** Caso d’uso Gestisci ordine

**Pre-Condizioni:**

* è presente un’istanza **ordine** di Ordine

**Post Condizioni:**

* è stata creata un’istanza **p** di Pagamento
* l’attributo “tipoPagamento" di **p** è stato aggiornato con *tipoPagamento*
* l’istanza **p** è stata associata ad **ordine**

**Contratto CO6: calcolaTotale**

**Operazione:** calcolaTotale(Ordine ordine)

**Riferimenti:** Caso d’uso Gestisci Ordine

**Pre-Condizioni:**

* è presente un’istanza **ordine** di **Ordine**
* è presente una **listaRigheDiOrdine[i]:RigaDiOrdine**
* è presente almeno un’istanza **rdo:RigaDiOrdine**

**Post Condizioni:**

* L’attributo “totale” di **ordine** viene aggiornato sommando i valori dell’attributo subtotale delle istanze **rdo** di **RigheDiOrdine**

**Progettazione**

In questa parte vengono illustrati i diagrammi di sequenza e il diagramma delle classi relativi a questa seconda iterazione.

**Diagrammi di sequenza**

Si tenga in considerazione che per la realizzazione dei seguenti diagrammi la classe **IdeaVerde** è stata utilizzata come **Controller** in accordo con l’omonimo pattern GRASP. La classe si occupa quindi di prendere i messaggi in arrivo dall’interfaccia e delegarli alle altre classi.

Immagine che contiene screenshot, mappa

Descrizione generata automaticamente

Figura 1 Crea un nuovo ordine

E’ stata scelta la classe Archivio come **Creator** perché si occupa di registrare gli Ordini. Stessa cosa vale per la classe Ordine la quale registra oggetti di RigaDiOrdine.

Immagine che contiene screenshot

Descrizione generata automaticamente

Figura 2 Seleziona una pianta

Immagine che contiene mappa, screenshot

Descrizione generata automaticamente

Figura 3 Aggiunge una pianta

Immagine che contiene screenshot

Descrizione generata automaticamente

Figura 4 Aggiorna la quantità delle piante in magazzino

Immagine che contiene screenshot

Descrizione generata automaticamente

Figura 5 Seleziona il tipo di spedizione

Immagine che contiene screenshot

Descrizione generata automaticamente

Figura 6 Calcola il totale dell'ordine

Immagine che contiene screenshot

Descrizione generata automaticamente

Figura 7 Conferma l'ordine

Il diagramma di sequenza relativo all’operazione selezionaPagamento(String tipoPagamento) è stato omesso poiché identico a quello di selezionaSpedizione(String tipoSpedizione) cambiando le classi opportune.

**Diagramma delle classi**

Per migliorare la leggibilità del diagramma delle classi è stato deciso di riportare all’interno della classe **IdeaVerde** i metodi relativi a questa iterazione. Inoltre, sono stati rimossi i metodi setter e getter.

Immagine che contiene screenshot

Descrizione generata automaticamente