



Taxi Service

Relazione progetto **Programmazione III e
Laboratorio di Programmazione III**
Prof. Angelo Ciaramella

a cura di
Salvatore Di Costanzo
Matricola: 0124001483

Descrizione Progetto

Il progetto prevede la realizzazione di un'applicazione per la gestione di Taxi.

È un progetto di tipo gestionale che permette agli utenti registrati sulla piattaforma di effettuare la prenotazione del servizio taxi, e , ad un utente amministratore del servizio Taxi di poter evadere o rifiutare le richieste.

Per chi è sviluppata l'applicazione?

Gli attori principali che avranno modo di interfacciarsi con l'applicazione sono:

1. **Passeggero:** Il quale potrà, dopo aver effettuato l'autenticazione, compilare un form contenenti le informazioni utili per la prenotazione del Taxi.
2. **Radio Taxi:** Il quale potrà, dopo aver effettuato l'autenticazione, evadere o rifiutare le richieste degli utenti.

Cosa ho usato?

- Il linguaggio di programmazione usato è Java
- Sono stati usati Design Pattern, di importanza fondamentale per lo sviluppo dell'applicazione
- Per le componenti grafiche è stato usato JavaFx con editor Scene Builder
- Per le componenti del DB è stato usato SQL Developer

Design Pattern

I Design Pattern usati per la realizzazione sono:

- Singleton
- Chain of Responsibility
- Strategy
- Vistor

Singleton

progetto.Connessione

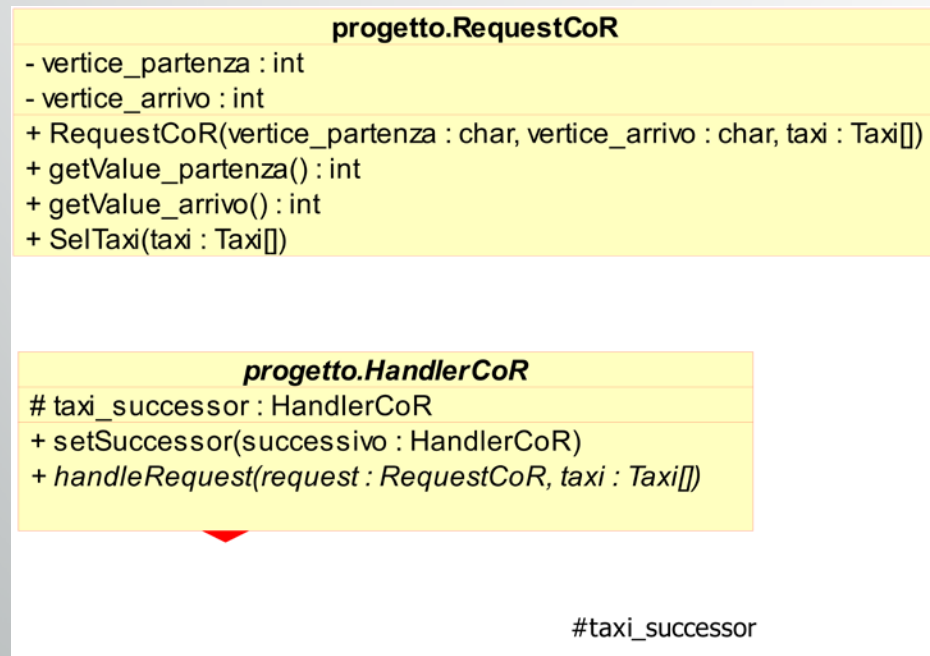
- instance : Connessione
- + connect : Connection
- Connessione()
- + getConnessione() : Connessione
- ~ createStatement() : Statement

Quando un particolare oggetto nel sistema ha una responsabilità su cui altri oggetti fanno affidamento bisogna identificarlo tra gli altri oggetti.

L'intento di Singleton è quello di assicurarsi che la classe abbia una sola istanza fornendo nel contempo un punto d'accesso globale.

Poiché il DB che usiamo è il medesimo per l'intero sistema usiamo Singleton per fornire accesso al database.

Chain of Responsibility



Tale pattern permette di separare il mittente di una richiesta dal destinatario in modo da consentire al più ad un oggetto di gestire le richieste.

Gli oggetti destinatari vengono organizzati in liste concatenate.

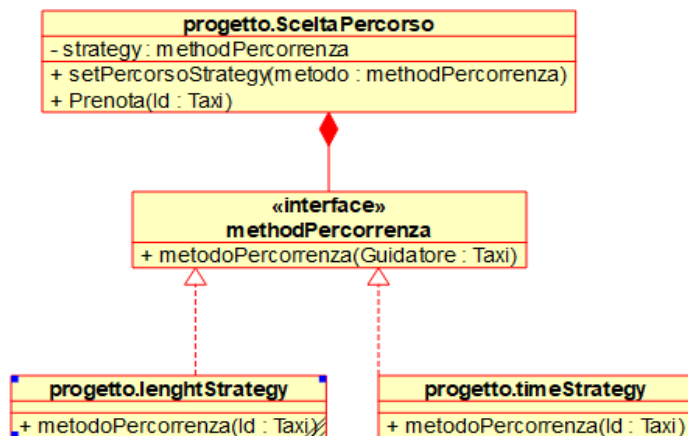
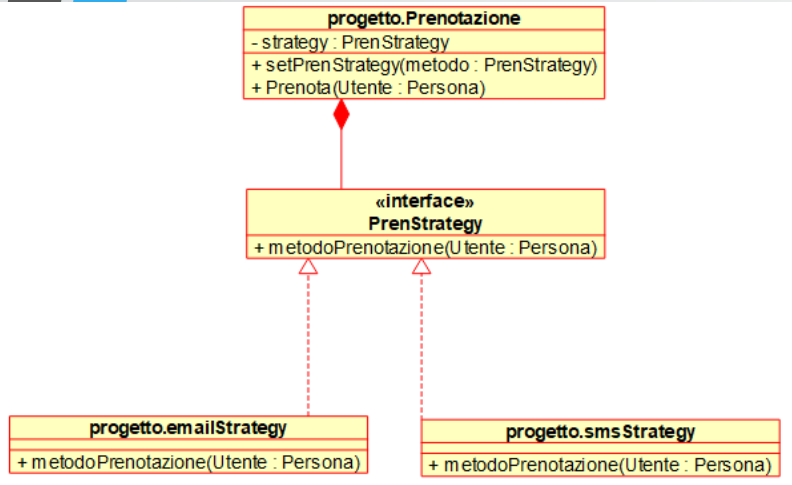
L'obiettivo del pattern è ridurre la dipendenza tra un oggetto che richiede un servizio e quello che riceve la chiamata, per mettendo a più oggetti di gestire la richiesta

Strategy

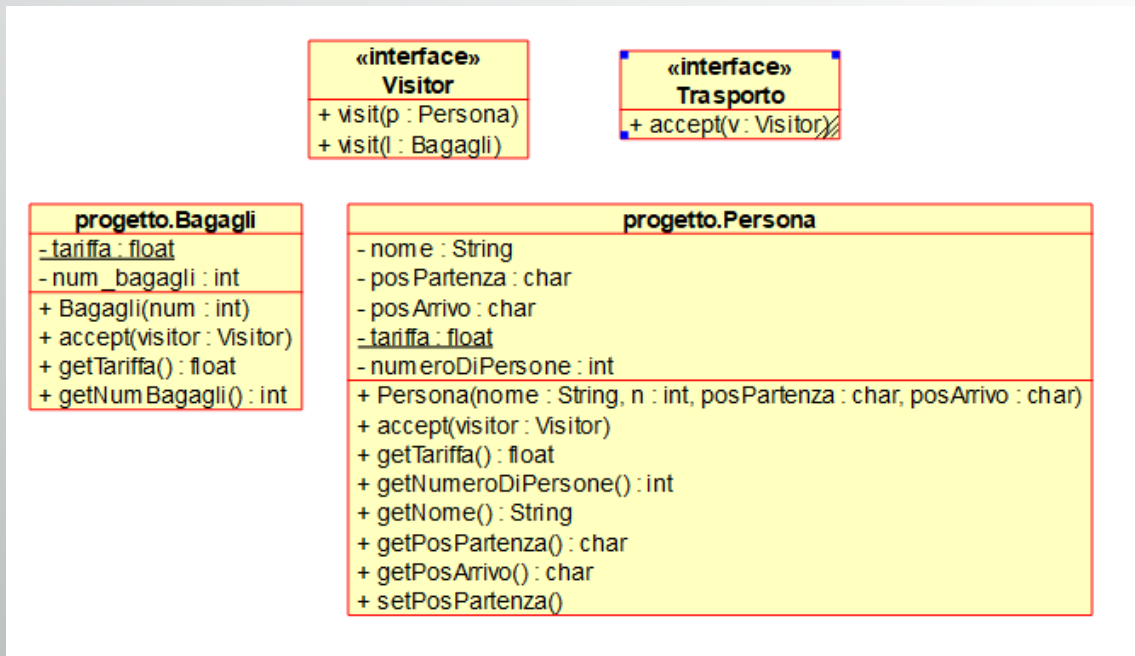
L'intento di Strategy è quello di incapsulare più strategie o approcci alternativi in classi separate, ognuna delle quali implementa un'operazione comune.

Il suo scopo quindi è definire una famiglia di algoritmi e renderli intercambiabili.

In questo progetto Strategy è stato usato sia per gestire il metodo di percorrenza che il **Taxi** vuole effettuare per portare il cliente a destinazione, sia per permettere al **Passeggero** di effettuare prenotazioni tramite SMS o EMail.



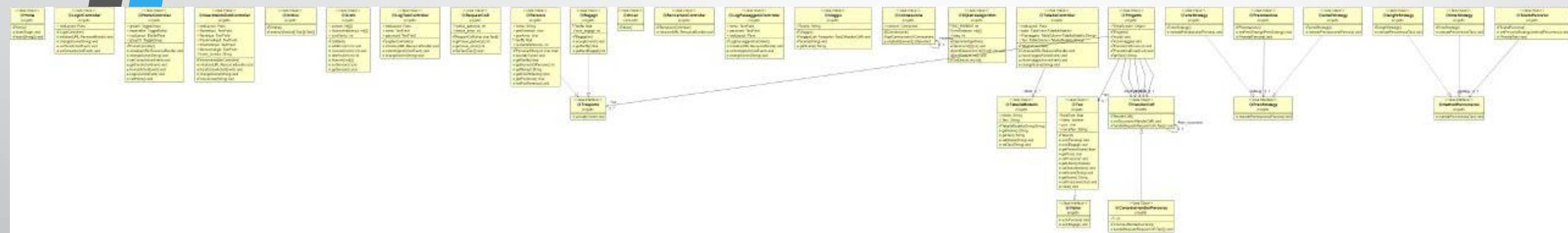
Vistor



Visitor è particolarmente utile quando una struttura di oggetti è costituita da classi con interfacce diverse ed è necessario che l'algoritmo esegua su ogni oggetto un'operazione differente.

In questo progetto Visitor è stato usato per trattare gli oggetti Bagagli e Persona in modo che entrambi potessero essere accettati sul Taxi

Unit 1: Introduction to the course	Unit 2: The history of the course	Unit 3: The development of the course	Unit 4: The future of the course	Unit 5: The impact of the course
<ul style="list-style-type: none"> What is the course? Why is it important? What are the goals? What are the topics? What are the methods? What are the resources? What are the challenges? What are the opportunities? What are the risks? What are the benefits? What are the costs? What are the results? What are the conclusions? What are the recommendations? What are the next steps? What are the final thoughts? 	<ul style="list-style-type: none"> What is the history of the course? What are the milestones? What are the key events? What are the major figures? What are the important dates? What are the significant achievements? What are the challenges faced? What are the opportunities created? What are the risks taken? What are the benefits realized? What are the costs incurred? What are the results achieved? What are the conclusions drawn? What are the recommendations made? What are the next steps planned? What are the final thoughts shared? 	<ul style="list-style-type: none"> What is the development of the course? What are the stages of growth? What are the key milestones? What are the major achievements? What are the challenges overcome? What are the opportunities seized? What are the risks managed? What are the benefits realized? What are the costs controlled? What are the results achieved? What are the conclusions drawn? What are the recommendations made? What are the next steps planned? What are the final thoughts shared? 	<ul style="list-style-type: none"> What is the future of the course? What are the goals for the future? What are the challenges ahead? What are the opportunities ahead? What are the risks ahead? What are the benefits ahead? What are the costs ahead? What are the results ahead? What are the conclusions ahead? What are the recommendations ahead? What are the next steps ahead? What are the final thoughts ahead? 	<ul style="list-style-type: none"> What is the impact of the course? What are the effects of the course? What are the consequences of the course? What are the outcomes of the course? What are the results of the course? What are the achievements of the course? What are the challenges of the course? What are the opportunities of the course? What are the risks of the course? What are the benefits of the course? What are the costs of the course? What are the results of the course? What are the conclusions of the course? What are the recommendations of the course? What are the next steps of the course? What are the final thoughts of the course?





Grazie per l'attenzione