

## Analisi della Coesione

### Classe Contatto (Alta Coesione)

- Rappresenta un singolo contatto con attributi strettamente correlati
- Incapsula le informazioni del contatto (cognome, nome, numeri di telefono, email)
- I metodi sono focalizzati sull'ottenere e impostare le proprietà del contatto
- Valutazione Coesione: Alta (Principio di Responsabilità Unica ben implementata)

### Classe ContattoComparator (Alta Coesione)

- Esclusivamente responsabile del confronto di oggetti Contatto
- Metodo singolo con un chiaro scopo di ordinare i contatti
- Valutazione Coesione: Alta

### Classe Rubrica (Coesione Moderata-Alta)

- Gestisce una lista di contatti
- Fornisce metodi per la gestione dei contatti (aggiunta, rimozione, modifica, ricerca)
- Delega le operazioni di file alla classe padre (InOutRubrica)
- Valutazione Coesione: Moderata-Alta

### Classe InOutRubrica (Alta Coesione)

- Completamente focalizzata sulle operazioni di input/output per i contatti
- Due metodi principali: salvataggio e caricamento delle liste di contatti
- Valutazione Coesione: Alta

### Classe RubricaGUI (Coesione Moderata)

- Gestisce operazioni relative all'interfaccia grafica
- Contiene multiple responsabilità:
  1. Creazione di componenti GUI
  2. Gestione delle interazioni utente
  3. Gestione delle operazioni sui contatti
- Valutazione Coesione: Moderata (Potrebbe essere migliorata suddividendo in classi più piccole)

## Analisi Accoppiamento

### Caratteristiche di Basso Accoppiamento:

1. **Iniezione Dipendenze:** Le classi sono debolmente accoppiate attraverso interfacce e composizione
2. **Dipendenze Dirette Minime:** La maggior parte delle classi interagisce attraverso metodi ben definiti
3. **Separazione delle Preoccupazioni:** Classi diverse gestiscono diversi aspetti del sistema di gestione dei contatti

## **Osservazioni specifiche sull'Accoppiamento:**

### **Accoppiamento Forte:**

- RubricaGUI è strettamente accoppiata con la classe Rubrica
- Dipendenza diretta tra Rubrica e Contatto
- InOutRubrica è accoppiata con Rubrica attraverso l'ereditarietà

### **Accoppiamento Debole:**

- ContattoComparator è debolmente accoppiata, dipendendo solo dall'interfaccia di Contatto
- La gestione dei file è astratta in InOutRubrica

## **Raccomandazioni per il Miglioramento:**

### **1. Riorganizzazione di RubricaGUI:**

- Suddividere in componenti più piccoli e focalizzati
- Separare la creazione dell'interfaccia, la gestione degli eventi e la logica di business
- Utilizzare la composizione invece di metodi estesi

### **2. Iniezione Dipendenze:**

- Considerare l'uso di interfacce per un disaccoppiamento migliore
- Implementare l'iniezione di dipendenza per Rubrica in RubricaGUI

### **3. Gestione degli Errori:**

- L'attuale registrazione degli errori potrebbe essere migliorata
- Considerare un'gestione **delle eccezioni personalizzata**

## **Metriche di Qualità del Codice:**

- Modularità: Buona
- Estensibilità: Moderata
- Manutenibilità: Buona
- Leggibilità: Alta

Il codice dimostra un design orientato agli oggetti solido con una buona separazione delle preoccupazioni e un accoppiamento relativamente basso. Il punto di forza principale risiede nelle chiare responsabilità di ogni classe e nella logica di gestione dei contatti ben incapsulata.