CAPOLVORO RUBRICA TELEFONICA

Il mio capolavoro è un programma Java che gestisce una rubrica telefonica, permettendo l'inserimento, la visualizzazione, la ricerca, la modifica e la cancellazione dei contatti. Inoltre, offre funzionalità come la ricarica del saldo telefonico e la simulazione di chiamate, con la possibilità di ordinare i contatti utilizzando diversi algoritmi di ordinamento.

**Il percorso che ho seguito**

1. **Definizione del problema e obiettivi**: All'inizio, ho identificato le funzionalità principali che il programma doveva offrire: gestione dei contatti, ricerca, modifica, cancellazione, ricarica del saldo e simulazione di chiamate. Ho anche deciso di aggiungere diversi algoritmi di ordinamento per migliorare l'organizzazione dei contatti.
2. **Progettazione del programma**: Ho iniziato a progettare la struttura del programma, definendo le classi e i metodi necessari. La classe principale, **Main**, contiene il metodo **main** e gestisce il menu di navigazione. La classe **Contatto** rappresenta i contatti nella rubrica, con attributi come nome, cognome, telefono e saldo. Inoltre, ho creato un enum **tipoContratto** per rappresentare i diversi tipi di contratto: abitazione, cellulare e aziendale.
3. **Implementazione delle funzionalità base**: Ho implementato le funzioni principali, come l'inserimento e la visualizzazione dei contatti. Per l'inserimento, ho creato un metodo **leggiPersona** che legge i dati di un contatto dall'utente. Per la visualizzazione, ho creato il metodo **visualizza** che mostra tutti i contatti presenti nella rubrica.
4. **Aggiunta delle funzionalità avanzate**: Successivamente, ho aggiunto le funzionalità di ricerca, modifica e cancellazione dei contatti. Ho implementato metodi come **ricerca**, **RicercaIndex** e **cancellazione** per gestire queste operazioni. Ho anche sviluppato la funzione per ricaricare il saldo telefonico e simulare chiamate.
5. **Implementazione degli algoritmi di ordinamento**: Ho implementato tre diversi algoritmi di ordinamento: selection sort, insertion sort e bubble sort. Questi algoritmi permettono di ordinare i contatti in base al cognome e, in caso di parità, in base al nome. Questo ha migliorato l'organizzazione dei contatti e ha reso più facile la loro gestione.
6. **Utilizzo della classe utility**: Ho creato una classe **utility** nel package **tools** che contiene metodi utili come **ClrScr** per pulire lo schermo, **Wait** per introdurre delle pause e **menu** per gestire la navigazione del menu principale. Questi metodi migliorano l'usabilità e la struttura del programma.
7. **Testing e debugging**: Durante l'intero processo di sviluppo, ho testato il programma per assicurarmi che tutte le operazioni funzionassero correttamente. Ho identificato e risolto vari bug, come errori nella gestione degli input e problemi di ordinamento.

**Riflettendo sull'esperienza**

Questa esperienza mi ha insegnato molto sull'importanza della pianificazione e della progettazione dettagliata. Ho imparato a suddividere un progetto complesso in parti più gestibili e a svilupparle una alla volta. Gli errori commessi mi hanno insegnato a essere paziente e attento ai dettagli, e a vedere i problemi come opportunità di apprendimento. In definitiva, questo progetto mi ha aiutato a diventare un programmatore più competente e consapevole.