**Relazione scritta  
PROGRAMMA GESTIONALE DI UNA CONCESSIONARIA DI AUTO  
Progetto di Informatica e Tecnologie di Progettazione di Sistemi Informatici e di Telecomunicazioni (TPSIT)  
Gruppo 1: Diego Albertin, Francesco Di Lena, Alessio Donini, Alex Niccolò Ferrari  
Classe 3a F   
A.S. 2020-2021   
I.T.I “Ferruccio Viola”**

**Premessa**

Prima di realizzare la scrittura del codice si è proceduto ad un’analisi: prima dei requisiti iniziali che avrebbe richiesto il software, poi funzionale per stabilire le funzioni del programma e, infine, una tecnica. Man mano che si è proceduto nella discussione fra i componenti del gruppo e alla realizzazione del programma, sono sorti degli aspetti che non erano stati preventivati inizialmente, oppure dei problemi di realizzazione che hanno portato a rivedere più volte i requisiti, le funzioni e la parte tecnica del programma. Le seguenti sono le analisi “finali” che consentono di comprendere a pieno tutti gli aspetti relativi al software.

**Analisi dei requisiti**

**La situazione iniziale**

Un’ azienda di automobili, Concessionaria Automotors S.N.C. di via Alcide de Gasperi 21 di Rovigo (fittizia), deve gestire il proprio parco auto. Vende auto nuove ed usate. Vende auto di diverse marche e diversi modelli, di diversi colori e di diverse motorizzazioni (cilindrata/potenza e carburante). Si rende necessario realizzare un software che possa far vedere quante auto sono in vendita attualmente, quante sono state vendute ed eventualmente filtrarle per una (o eventualmente più di una) delle caratteristiche. Le caratteristiche, per ogni tipo di auto sono state ben definite.

* **Auto nuove in vendita**: marca, modello, alimentazione, cilindrata, potenza, anno di produzione, colore, prezzo.
* **Auto usate in vendita**: marca, modello, alimentazione, cilindrata, potenza, anno di produzione, colore, targa, numero di proprietari, chilometri percorsi e prezzo.
* **Auto vendute nuove**: marca, modello, alimentazione, cilindrata, potenza, anno di produzione, colore, targa, acquirente, data di acquisto, prezzo di vendita iniziale, prezzo di vendita finale.
* **Auto vendute usate**: marca, modello, alimentazione, cilindrata, potenza, anno di produzione, colore, targa, numero di proprietari, chilometri percorsi, acquirente, data di acquisto, prezzo di vendita iniziale, prezzo di vendita finale.

Oltre a visualizzare le automobili già presenti nella concessionaria deve essere possibile inserire delle nuove automobili nell’elenco, oppure eliminarle e spostarle, in modo da tenerlo sempre aggiornato.  
L’azienda ha un’unica sede ed è di dimensioni piccole, per cui le auto da gestire non possono superare le migliaia.

**Chi deve usare il programma?**

I destinatari del programma sono i dipendenti della concessionaria, non i clienti, che necessitano di conoscere il proprio parco di automobili in vendita e vendute e che hanno bisogno di avere a disposizione sempre le istruzioni del programma nel computer per conoscere sempre le operazioni da compiere e i comandi da utilizzare.

**Quali requisiti devono essere soddisfatti per far funzionare e rendere efficiente l’utilizzo del programma?**

La concessionaria deve essere fondamentalmente dotata di sistemi informatici che consentano l’esecuzione del programma, così come di infrastrutture di rete, che permettano sempre lo scambio dell’elenco di automobili in vendita e vendute tra tutti i computer presenti nella concessionaria, in modo che qualsiasi consulente e dipendente possa aver accesso ai dati.   
Questa ultima parte, però, insieme ad altri requisiti che sono estremamente necessari per l’intera gestione della concessionaria, non vengono considerati per la creazione del programma previsto dal progetto collaborativo, ma posti come una base per il suo funzionamento che è già presente. Il progetto si è quindi concentrato sulla parte software e non su quella hardware.

**Tempo di realizzazione del progetto**

Le consegne inizialmente poste dal cliente sono state le seguenti:

* analisi dei requisiti 05-12-2020;
* analisi funzionale 12-12-2020;
* analisi tecnica 19-12-2020;
* codice sorgente commentato adeguatamente 27-01-2021;
* relazione finale 27-01-2021;
* video-relazione finale 30-01-2021.

Il termine fissato è stato poi prorogato a febbraio, e, infine, al 31 marzo 2021. Il tempo di realizzazione del progetto fornito dal cliente è stato quindi di quattro mesi: dicembre 2020, gennaio-febbraio-marzo 2021.

**Analisi funzionale**

Il programma ha un interfaccia di tipo riga di comando in cui vengono inseriti i diversi parametri richiesti per la visualizzazione dell’elenco.

* All’inizio il programma si apre con una breve schermata introduttiva che esplica alcune funzionalità del programma stesso, indica la data del giorno corrente.
* Al primo avvio del programma, oppure in ogni caso in cui non riuscisse a trovare i file delle auto, viene visualizzato un messaggio di errore: la cartella e i file che contengono le auto non sono presenti all’interno del disco rigido, per cui è necessario crearli. L’utente può decidere di crearli inserendo la parola chiave “**SÌ**”, oppure può saltare la creazione inserendo “**NO**”: quest’ultima opzione è stata decisa perché, nel caso in cui nei successivi funzionamenti del programma, l’utente sposti i file necessari, può chiudere il programma e reinserirli manualmente. Nel caso in cui, invece, la cartella sia stata creata ma i file no, le richieste di creazione dei file sono quattro: una per creare il file con le auto nuove in vendita, il secondo le auto usate in vendita, il terzo le auto nuove vendute, il quarto le auto usate vendute.
* Una volta effettuata la creazione della cartella e dei file, se necessario, il programma effettua la sua prima vera richiesta: si vuole fare una ricerca fra le auto, inserirne, eliminarne o spostarne una. L’utente decide inserendo le parole chiave che corrispondono al nome della funzione: “**RICERCA**”, “**INSERISCI**”, “**ELIMINA**”, oppure “**SPOSTA**”. Una quinta funzione è poi disponibile: nel caso in cui l’utente voglia leggere il manuale del programma, per impararne l’utilizzo, può inserire la parola chiave “**AIUTO**”.

**Prima funzione: RICERCA**

Una volta inserita l’apposita parola chiave, si accede alla funzione ricerca. Questo tipo di funzione, insieme a quella di spostamento e eliminazione, può essere svolta solamente in caso in cui i file contengano delle auto. La prima richiesta è quella se si vuole accedere alle auto in vendita o vendute: la scelta è obbligata e può avvenire inserendo rispettivamente le parole chiave “**VENDITA**” oppure “**VENDUTA**”. La seconda richiesta, poi, riguarda le auto nuove o usate, dove le possibilità di scelta sono due, anche qui obbligata: la parola chiave può essere “**NUOVA**”, “**USATA**”. Dopo queste prime richieste si possono presentare due scenari: se non ci sono auto all’interno del file corrispondente al tipo scelto allora la ricerca si blocca e dà un messaggio di errore, invitando l’utente ad usare la funzione di inserimento di una nuova auto prima di eseguire la ricerca; se ci sono delle auto presenti nei file, allora comincia a chiedere le caratteristiche delle auto, a partire dalla marca, poi tutte le altre caratteristiche, che variano in base al tipo di auto che si vuole ricercare. L’inserimento di tutte le caratteristiche non è necessario: infatti, se si vuole saltare una o più caratteristiche, senza però terminare la richiesta, basta non inserire nulla e premere invio; nel caso, invece, in cui si voglia terminare la ricerca ad una certa caratteristica, si preme invio per arrivare a quella successiva e poi, in quest’ultima, si inserisce la tripla barra “ **///** ”; se si inserisce la tripla barra alla prima caratteristica (la marca) viene visualizzato l’elenco completo delle auto appartenenti alla categoria scelta inizialmente .   
Una volta terminata la richiesta di filtri per la ricerca, mostra i risultati: qui si possono presentare due casi. Il primo è quello in cui non è stato trovato nulla, quindi il programma invita l’utente a ripetere la ricerca con parametri diversi, mentre nel secondo caso la ricerca mostra dei risultati: innanzitutto mostra quanti risultati sono stati ottenuti, poi mostra quali caratteristiche, in serie, saranno visualizzate in base al tipo di auto ricercata e, infine, mostra le auto, in ordine numerato e alfabetico. L’utente può visionare quindi le varie caratteristiche delle auto, e, dopo che è stato raggiunto questo obbiettivo, può decidere se compiere altre azioni oppure no: il programma chiede se lo si vuole riavviare per eseguire ulteriori operazioni (che possono essere sempre ricerca, oppure delle altre) oppure se lo si vuole chiudere. Se l’utente a questa richiesta sceglie “**SÌ**”, allora il programma viene riportato alla schermata iniziale, altrimenti, se l’utente inserisce “**NO**”, il programma si chiude automaticamente.

**Seconda funzione: INSERIMENTO**

La seconda funzione utilizzabile è l’inserimento, accessibile attraverso l’omonima parola chiave. Essa consiste nell’inserire all’interno dei file una nuova auto, quindi, a differenza delle altre funzioni, questa può essere eseguita anche nel caso in cui i file non contenessero auto. Le prima richiesta, come avviene nella ricerca, è di scegliere se si vuole inserire un’auto in vendita o venduta, così come la seconda, in cui la decisione avviene tra nuova e usata; poi avviene la richiesta di tutte le caratteristiche dell’auto che si deve inserire in base alle prime due scelte: qui però, a differenza della ricerca, l’inserimento delle caratteristiche è obbligatorio, quindi se si lasciano spazi vuoti il programma darà errore. Una volta terminato l’inserimento da tastiera di tutte le caratteristiche della macchina, il programma le riepiloga automaticamente, in modo che l’utente si possa accorgere di eventuali errori di inserimento, e chiede se si vuole procedere con il salvataggio delle modifiche: se non ci sono errori e decide di farlo, l’utente inserisce “**SÌ**” alla richiesta e il programma visualizza “**Salvataggio in corso...**”, invitandolo a non chiudere il programma, e poi, se non ci sono stati problemi, visualizza “**Salvataggio completato con successo**”. Nel caso in cui ci siano dei problemi, però, visualizza un messaggio di errore: esegue dei nuovi tentativi (al massimo 3), ma se ancora non riesce invita l’utente a chiudere e riavviare il programma, oppure a contattare l’amministratore in caso di problemi di autorizzazione.  
Ritornando comunque alla scelta se salvare o meno, se l’utente inserisse “**NO**” alla richiesta , allora il programma chiederebbe anche qui se volesse riavviare il programma oppure no: questo quesito viene posto anche al termine del salvataggio, ma in modo particolare se l’utilizzatore del software non volesse salvare, allora potrebbe ripetere l’inserimento dopo essere ritornato alla schermata iniziale, oppure passare ad un’altra funzione, oppure, ancora, chiudere il programma.

**Terza funzione: ELIMINAZIONE**

Alla funzione di eliminazione di può accedere con la giusta parola chiave e consiste nell’eliminare definitivamente un’automobile da un certo file, in base al tipo di auto che viene selezionata. Dal punto di vista puramente funzionale essa è in realtà costituita da due funzioni: dalla ricerca e dall’eliminazione stessa. La prima parte, quindi, funziona come la ricerca: si inseriscono le caratteristiche obbligatorie, quali la vendita/venduta e nuova/usata, e, se si vuole, si inseriscono le altre caratteristiche delle auto. Una volta mostrati i risultati della ricerca, quindi con l’elenco numerato, il programma chiede di inserire il numero corrispondente all’auto che si desidera eliminare: l’utente può inserire solamente i numeri che sono presenti all’interno dell’elenco, altrimenti il programma dà un messaggio di errore e chiede di nuovo l’inserimento di un parametro corretto. Una volta inserito il numero e premuto invio, il programma, come avviene nell’inserimento, mostra di nuovo le caratteristiche dell’auto che si è scelta, in modo che l’utente possa verificare se ha fatto la scelta giusta. Anche qui se la scelta è “**SÌ**”, allora il programma prosegue con la vera eliminazione, con la quale avviene salvataggio delle modiche e se ci sono degli errori lo segnala, altrimenti se il salvataggio è terminato, oppure se la scelta è stata “**NO**” chiede se si vuole proseguire con un riavvio del programma, oppure una chiusura.

**Quarta funzione: SPOSTAMENTO**

Con la parola “**SPOSTA**” si accede allo spostamento, funzione, anche questa, costituita da più funzioni che lavorano insieme: ricerca, spostamento stesso, inserimento ed eliminazione. Essa consiste nell’eliminare una macchina che si trova all’interno di un file, per inserirla all’interno di un file diverso, quindi spostandola, per esempio, da in vendita nuova a venduta nuova. Le prime richieste da parte del programma sono quelle della ricerca, poi, una volta visualizzato l’elenco numerato, l’utente decide quale auto vuole spostare inserendo il numero, poi la richiesta è sempre se è sicuro di voler spostare l’auto inserita con il numero: se la risposta è “**SÌ**”, allora il programma inizia a richiedere dove si vuole trasferire l’auto. Si chiede sempre se si vuole spostare a vendita/venduta e a nuova/usata, sempre utilizzando le parole chiave “**VENDITA**”/”**VENDUTA**” e “**NUOVA**”/”**USATA**”. Si possono presentare tre casi. Il primo è quando l’utente decide di spostare l’auto nello stesso elenco di provenienza: ovviamente non è uno spostamento possibile, quindi il software mostra un messaggio di errore. Il secondo caso è quando l’utente inserisce uno spostamento che è possibile ma che deve avvenire da un elenco con meno informazioni a uno che ne ha di più, come da in vendita nuova a venduta nuova: si rende necessario quindi l’inserimento delle caratteristiche mancanti, quindi il programma richiede l’inserimento delle caratteristiche mancanti. Il terzo caso, infine, avviene quando lo spostamento è valido e deve avvenire da un elenco più grande ad uno più piccolo: in questo caso bisogna eliminare le informazioni che non sono più necessarie, operazione a cui provvede in automatico il software.  
Negli ultimi due casi, se tutte le operazioni vengono compiute, il programma mostra di nuovo l’auto con le nuove caratteristiche inserite o eliminate e chiede se si vuole procedere al salvataggio: se l’utente inserisce “**SÌ**", allora avviene prima l’inserimento dell’auto con le nuove informazioni all’interno del file in cui si è deciso di inserirla, poi esegue l’eliminazione nel file dove si trovava prima. Anche qui nel caso in cui ci siano errori vengono segnalati all’utente durante l’esecuzione di entrambe le fasi di salvataggio.  
Se, ritornando al primo caso spiegato prima in cui lo spostamento inserito dall’utente non sia valido, oppure se l’utente decide di non salvare le modifiche, il software chiede all’utente se desidera riavviare il programma per eseguire ulteriori operazioni, oppure chiuderlo.

**Quinta funzione: AIUTO**

La quinta funzione, l’aiuto, non esegue particolari operazioni, ma mostra solamente il manuale d’uso del programma nel caso in cui l’utilizzatore del programma ne abbia bisogno. Il prelevamento da file delle istruzioni avviene solo nel momento in cui si attiva questa funzione, non all’inizio del programma, così se ci sono degli errori di lettura del file vengono visualizzati in quel momento: la gestione degli errori è, anche in questa parte di programma, simile a quella presentata nelle altre funzioni.

**Analisi tecnica**

Il software per la gestione della concessionaria di auto è stato realizzato attraverso il linguaggio di programmazione   
**C# Console**, quindi l’interfaccia usata è una **CLI (Command Line Interface)**. L’editor di sviluppo utilizzato è stato   
**Visual Studio** **2017**, e si è utilizzato **Microsoft .NET Core**.

**Direttive USING utilizzate**

Per realizzare il codice sono state utilizzate diverse direttive **using**:

* ***using.System***, che permette l’uso di metodi generali come ***Console.WriteLine()***;
* ***using System.IO***, che permette l’uso di metodi per l’input/output per i file, come ***File.ReadAllText()***;
* ***using System.Globalization***, che permette l’uso di metodi che consentono il confronto di stringhe secondo delle ben precise regole linguistiche;
* ***using System.Threading.Tasks*,** chepermette l’uso di metodi che posticipano l’avvio delle istruzioni, come ***Task.Delay().Wait().***

**Parti di divisione del codice**

Il codice può essere fondamentalmente diviso in due parti:

* inizializzazione e dichiarazione di variabili globali necessarie in tutte le funzioni oltre al ***Main***;
* le funzioni che svolgono diverse operazioni separatamente.

**Variabili inizializzate e dichiarate**

Nella prima parte del codice sono state dichiarati diversi tipi di variabili: **intere**, **stringa**, **carattere**, **booleane**, **vettori unidimensionali di stringhe** e **vettori multidimensionali di stringhe**. Per poter essere utilizzate anche nelle funzioni, esse fanno utilizzo del modificatore ***static***. Per maggiori informazioni sui nomi delle variabili e il loro compito, vedere la documentazione interna del software (commenti).

**Le funzioni**

Sono dodici le funzioni locali che si trovano all’interno del programma:

* funzione ***Main***;
* funzione ***ControlloPrimaScelta***;
* funzione ***ControlloSecondaScelta***;
* funzione ***SceltaMatrici***;
* funzione ***RicercaFile***;
* funzione ***InserisciFile***;
* funzione ***EliminaFile***;
* funzione ***SpostaFile***;
* funzione ***Salvataggio***;
* funzione ***FineProgramma***;
* funzione ***ControlloRisposta***;
* funzione ***ChiusuraProgramma***.

***Main***

La funzione ***Main*** è la principale del programma. Viene avviata subito con l’avvio del software e si compone essenzialmente di sette parti:

* la visualizzazione della schermata di benvenuto, con la data, grazie al metodo ***DateTime.Now.ToString("dd/MM/yyyy")***, che la mostra quindi nel formato giorno/mese/anno;
* l’acquisizione del contenuto delle auto dai quattro file salvati in ***C:\Programma gestionale concessionaria*** (**arraynuovoinvendita.txt**, **arrayusatoinvendita.txt**, **arraynuovovenduto.txt**, **arrayusatovenduto.txt**), attraverso l’utilizzo del metodo ***File.ReadAllText()*,** che permette di leggere tutto il contenuto di un file testo e di salvarlo in un’unica stringa: le eccezioni che può generare sono ***FileNotFoundException***, ***DirectoryNotFoundException***, ***IOException*** e ***UnauthorizedAccessException***, che vengono tutte gestite attraverso l’uso di più blocchi ***try-catch***; poi procede alla divisione delle varie caratteristiche delle auto attraverso il metodo ***String.Split()***, dove la divisione fra le caratteristiche vengono riconosciute dall’uso del carattere due punti ‘**:**’, poi vengono prese e inserite all’interno di un array monodimensionale;
* l’inserimento delle auto all’interno di quattro diversi vettori multidimensionali, con ogni colonna che rappresenta una caratteristica, mentre ogni riga rappresenta un’unica auto;
* le richieste iniziali di inserimento dei parametri, in cui, con la lettura da tastiera, si usa il metodo ***.ToUpper***, in modo che non ci siano errori dovuti a maiuscole e minuscole e dove si rivolge, più volte, a ***ControlloPrimaScelta*** e ***ControlloSecondaScelta***;
* il riferimento prima alla funzione ***SceltaMatrici***, la quale si occupa del riferimento alle funzioni specifiche (***RicercaFile***, ***InserisciFile***, ***EliminaFile***, ***SpostaFile***);
* il codice che preleva le istruzioni d’uso del programma dal file **aiuto.txt**, sempre dalla stessa cartella e con il metodo ***File.ReadAllText()***, si occupa della gestione delle eccezioni e le visualizza a schermo il manuale;
* una volta terminato il resto delle operazioni, si rivolge alla funzione ***FineProgramma***.

Altri metodi presenti degni di nota sono:

* ***Console.SetWindowSize()***, che permette di gestire la grandezza della finestra del programma quando viene avviato;
* ***Console.Clear()***, che pulisce il contenuto della console ogni volta che il software viene riavviato;
* ***Console.BackgroundColor***, insieme a **ConsoleColor**, che permette di cambiare il colore allo sfondo della console;
* ***Console.ForegroundColor***, insieme a **ConsoleColor**, che permette di cambiare il colore dei caratteri della console.

Altre informazioni che spiegano il funzionamento del ***Main***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***ControlloPrimaScelta***

La funzione ***ControlloPrimaScelta*** viene richiamata dal ***Main*** nel caso in cui ci sia da verificare la correttezza dei parametri della prima scelta, cioè la scelta se accedere alle funzioni con auto in vendita o vendute, inseriti dall’utente da tastiera. Questa funzione è composta da un ciclo ***while*** attivato in caso di errore nell’inserimento ed è stata isolato qui perché viene utilizzato più volte, in modo da non riscrivere più volte lo stesso codice.

**ControlloSecondaScelta**

Anche la funzione ***ControlloSecondaScelta*** viene richiamata dal ***Main*** nel caso in cui ci sia da verificare la correttezza dei parametri della seconda scelta, cioè la scelta se accedere alle funzioni con auto nuove o usate, inseriti dall’utente da tastiera. Anche questa funzione è composta da un ciclo ***while*** attivato in caso di errore nell’inserimento ed è stata isolato qui perché viene utilizzato più volte, in modo da non riscrivere più volte lo stesso codice.

**SceltaMatrici**

La funzione ***SceltaMatrici*** si rende fondamentale perché permette di inserire le informazioni di variabili e matrici specifiche all’interno di variabili e matrici generali, che possono essere utilizzate all’interno delle funzioni specifiche (ricerca, inserimento, eliminazione, spostamento e salvataggio), con un notevole risparmio di codice. Si compone di due parti: una parte in cui, in base alle due scelte iniziali ***primaScelta*** e ***secondaScelta***, viene scelto quali variabili devono essere inserite all’interno di quelle generali, e un’altra parte dove, in base al valore di ***sceltaIniziale***, fa riferimento alle funzioni specifiche che sono state scelte con la prima richiesta. Altre informazioni che spiegano il funzionamento di ***SceltaMatrici***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***RicercaFile***

La funzione ***RicercaFile*** viene utilizzata nel caso in cui le intenzioni iniziali dell’utente possano essere la ricerca vera e propria, l’eliminazione e lo spostamento: questi ultimi due, in particolare, come detto all’interno dell’analisi funzionale, funzionano prima con una ricerca e poi con le funzioni specifiche. Detto questo, la funzione ***SceltaMatrici*** da riferimento alla funzione ***RicercaFile*** se il valore di ***sceltaIniziale*** è “**RICERCA**”, “**ELIMINA**”, oppure “**SPOSTA**”, con conseguente cambio delle richieste per l’utente visualizzate una volta che il funzionamento del programma. La divisione di questa funzione, per quanto riguarda il codice, può essere di tre parti:

* le richieste dei filtri di ricerca, che vengono sempre convertiti in maiuscolo con ***.ToUpper()***;
* la verifica di compatibilità dei filtri inseriti con ciò che si trova negli array, che avviene attraverso un confronto fra stringhe che, grazie alle impostazioni della cultura (con l’uso della classe ***CultureInfo*** e poi del metodo ***String.Compare*** insieme a ***Compare.Options.IgnoreSymbols***, che ignora i simboli), non tengono conto dei caratteri speciali;
* la visualizzazione a schermo dei risultati ottenuti, ordinati con un numero identificativo a lato.

Altre informazioni che spiegano il funzionamento di ***RicercaFile***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***InserisciFile***

Per effettuare l’inserimento all’interno dei file si usa la funzione ***InserisciFile***. Prima di accedere a questa funzione, però, le richieste delle caratteristiche delle auto, convertite tutte in maiuscolo con il metodo ***.ToUpper()***, si trovano all’interno della funzione ***SceltaMatrici***, che poi fa riferimento a ***InserisciFile*** nel caso in cui si decida per il salvataggio. All’interno di quest’ultima funzione, quindi, si inserisce la nuova automobile all’interno di un nuovo array unidimensionale, insieme alle altre auto e il carattere divisore del file (‘:’, l’array è ***autoDaInserire***, con numero di celle pari al numero delle auto esistenti più quella nuova) e si effettua l’ordinamento: attraverso un ciclo for, si confrontano le varie celle attraverso il confronto fra stringhe (sempre con il metodo ***string***.***Compare*** con prima la classe ***CultureInfo***, poi il metodo ***Compare.Options.IgnoreSymbols***, che ignora i simboli) e si ordinano con algoritmo paragonabile a ***Bubble Sort*** (le auto “salgono” dal basso nell’array come bolle nel bicchiere). Infine inserisce tutto il contenuto ordinato all’interno di una nuova stringa (***autoDaSalvare***), insieme alla quale fa riferimento alla funzione ***Salvataggio***. Altre informazioni che spiegano il funzionamento di ***InserisciFile***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***EliminaFle***

Dopo le scelta iniziali effettuate a inizio programma per eliminare un’auto e l’attivazione della funzione di ricerca, si arriva alla funzione ***EliminaFile***, alla quale ha fatto riferimento ***RicercaFile***. Durante la ricerca vengono salvati all’interno di un array (***posizioniAuto***) le posizioni delle auto ricercate; la funzione di eliminazione utilizza questa matrice per chiedere all’utente quale auto vuole eliminare: dopo che l’utente inserisce il numero, questo viene decrementato (le matrici hanno come primo valore 0) e viene utilizzato come indice per indentificare l’automobile da eliminare. In questo modo, viene creata una nuova stringa (***autoDaSalvare***) in cui vengono inserite tutte le auto, con le caratteristiche separate dai due punti, tranne quella con l’indice indicato dall’utente. Una volta terminata questa operazione, la funzione ***EliminaFile*** chiama ***Salvataggio***. Altre informazioni che spiegano il funzionamento di ***EliminaFile***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***SpostaFile***

Per eseguire lo spostamento, si fa uso delle funzioni ***RicercaFile***, ***SpostaFile***, ***InserisciFile*** e ***EliminaFile***. All’inizio si effettua la ricerca (***RicercaFile***) fra le auto, viene chiesto all’utente quale auto vuole spostare: in questo momento la funzione in esecuzione è ***EliminaFile***, la quale, sapendo che la scelta iniziale è “**SPOSTA**”, richiama ***SpostaFile***: qui sono contenute le due nuove richieste che riguardano la nuova destinazione dell’auto (vendita/venduta, nuova/usata), poi tutte le scelte ***if/else*** con le operazioni . Poi viene richiamata la funzione ***InserisciFile***, con conseguente inserimento dell’auto con le informazioni aggiornate all’interno del nuovo file e, infine, ritorna alla funzione ***EliminaFile*** per terminare il procedimento con l’eliminazione dal precedente file di appartenenza.Altre informazioni che spiegano il funzionamento di ***SpostaFile***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***Salvataggio***

La funzione ***Salvataggio*** è chiamata da ogni funzione che abbia bisogno del salvataggio dei dati nei file (inserimento, eliminazione, spostamento). In base alle scelte iniziali, salva le nuove informazioni nel file adeguato attraverso il metodo ***File.WriteAllText()***, che permette di cancellare il contenuto corrente di un file e salvare quello nuovo; in più permette, in caso di eliminazione del file coinvolto, di ricrearne uno nuovo con le nuove informazioni. Le eccezioni, ovvero ***DirectoryNotFoundException*** e ***IOException***, sono gestite attraverso l’uso di blocchi ***try-catch***.  
Altre informazioni che spiegano il funzionamento di ***Salvataggio***, delle sue istruzioni e dei suoi metodi si possono consultare nella documentazione interna del software.

***FineProgramma***

La funzione ***FineProgramma*** viene richiamata ogni qualvolta si arrivi al termine di un’operazione: in caso di scelte negative oppure quando ogni operazione possibile è terminata. Contiene una sola richiesta: il programma chiede all’utente se voglia uscire dal programma, oppure riavviarlo. Nel primo caso richiama la funzione ***ChiusuraProgramma***. Nel secondo caso, invece, ritorna al ***Main***, dove il programma si riavvia con la nuova esecuzione del ciclo ***do-while***.

***ControlloRisposta***

La funzione ***ControlloRisposta*** viene chiamata quando è necessario effettuare la verifica di un ciclo while dopo l’inserimento, da tastiera dell’utente, di una risposta che chiede se procedere con delle operazioni oppure no, come nella funzione ***FineProgramma***: qui le uniche due risposte possibili sono “**SÌ**”, oppure “**NO**”, quindi tutte le altre non sono valide.

***ChiusuraProgramma***

La funzione ***ChiusuraProgramma*** viene attivata nei casi in cui sia necessario uscire dal programma, quindi quando si generano delle eccezioni, oppure dopo la funzione ***FineProgramma*** : contiene imetodi ***Task.Delay().Wait()*** per far aspettare qualche secondo prima dell’uscita in modo che l’utente possa rendersene conto e ***Enviroment.Exit()***, dove esce automaticamente dal programma.

**Lista di parole chiave usate nel programma**

* **SÌ**: risposta utilizzata in varie richieste, come nelle richieste delle eccezioni, oppure alla fine del programma.
* **NO**: risposta utilizzata in varie richieste, come nelle richieste delle eccezioni, oppure alla fine del programma.
* **RICERCA**: utilizzabile nella scelta iniziale per accedere alla funzione di ricerca.
* **INSERISCI**: utilizzabile nella scelta iniziale per accedere alla funzione di inserimento.
* **ELIMINA**: utilizzabile nella scelta iniziale per accedere alla funzione di eliminazione.
* **SPOSTA**: utilizzabile nella scelta iniziale per accedere alla funzione di spostamento.
* **AIUTO**: utilizzabile nella scelta iniziale per accedere alla funzione di aiuto.
* **HELP**: parametro alternativo utilizzabile nella scelta iniziale per accedere alla funzione di aiuto.
* **?** : parametro alternativo utilizzabile nella scelta iniziale per accedere alla funzione di aiuto.
* **VENDITA**: utilizzabile per accedere alle auto in vendita nella funzione di ricerca, inserimento, eliminazione e in quella di spostamento.
* **VENDUTA**: utilizzabile per accedere alle auto vendute nella funzione di ricerca, inserimento, eliminazione e in quella di spostamento.
* **NUOVA**: utilizzabile per accedere alle auto nuove nella funzione di ricerca, inserimento, eliminazione e in quella di spostamento.
* **USATA**: utilizzabile per accedere alle auto usate nella funzione di ricerca, inserimento, eliminazione e in quella di spostamento.
* Numeri **1,2,3,4,…** : numeri da inserire nella funzione di eliminazione e spostamento, che variano in base alle auto presenti nei file selezionati.
* **Caratteristiche delle auto**: sono le diverse caratteristiche che riguardano un’auto e possono essere inserite nelle funzioni di ricerca, inserimento, eliminazione e spostamento.

**Lavoro di gruppo**

**Ruoli e lavoro svolto**

Per realizzare il progetto, ogni componente del gruppo ha svolto una parte del lavoro e ha avuto un ruolo.

* **Diego Albertin**: ha assunto il ruolo di Git Master, occupandosi della gestione della repository di Git in cui è stato conservato il programma e dove sono state effettuate le unioni delle parti di codici prodotte dai componenti del gruppo; ha contribuito alla realizzazione della funzione di ricerca e di eliminazione, oltre agli effetti grafici della console.
* **Francesco Di Lena**: si è occupato della realizzazione delle funzioni di lettura e scrittura da file, di ricerca e di inserimento; ha contribuito alla realizzazione della funzione di eliminazione.
* **Alessio Donini**: si è occupato della gestione delle eccezioni durante le fasi di lettura e scrittura da file; ha contribuito al miglioramento della funzione di ricerca, così come della funzione di inserimento.
* **Alex Niccolò Ferrari**: ha assunto il ruolo di project manager, occupandosi di dirigere il lavoro di tutto il gruppo; si è occupato della realizzazione della funzione di spostamento e di eliminazione, ma anche della riorganizzazione del codice per un maggior ordine.

**Tempi di lavoro**

Il progetto è iniziato a dicembre 2020, quando è iniziata la discussione sui requisiti del programma, il suo funzionamento e la sua strutturazione tecnica, con conseguente produzione delle varie analisi. Per la scrittura del codice si è iniziato a gennaio 2021, per terminare a metà marzo. Durante questa fase, durante la scrittura di ogni funzionalità, si è proceduto all’esecuzione di test per verificare la piena funzionalità del programma. La relazione è stata, invece, scritta a partire da inizio marzo.

**Conclusioni**

Il lavoro ha portato alla realizzazione di un software gestionale funzionante in ogni sua parte, che non riscontra problemi nell’utilizzo e non genera alcuna eccezione non gestita: lo scopo iniziale è stato quindi ampiamente raggiunto.  
I tempi di consegna sono stati poi rispettati e, per quanto riguarda il gruppo, c’è stata una buona suddivisione del lavoro e delle responsabilità che ha consentito di sviluppare agevolmente il software.