RestaurantManager

Bella(1942308), De Persio(1938022)

1 Introduzione

Questo progetto da noi sviluppato consiste in un applicativo per la gestione di un ristorante che permette ai vari dipendenti di svolgere le loro mansioni principali e di scambiarsi informazioni utili a quest'ultime.
Il lavoro ha subito varie ripartizioni a causa dell'abbandono di un nostro collega ma abbiamo comunque diviso lo sviluppo ordinatamente.

Il progetto è stato ripartito nel modo seguente:

- Lo sviluppo della GUI
- Sviluppo backend

2 Descrizione delle Classi

2.1 Interfaccia Grafica

L'interfaccia grafica e' stata sviuppata utilizzando un plugin chiamato "WindowBuilder"

- **MenuGUI**: Schermata principale dalla quale l'utente può visionare il menu, scegliere se accedere ad una delle sezioni presenti o se chiudere il programma.
- **ChefGUI**: Sezione dedicata allo Chef nella quale è possibile aggiungere o rimuovere un piatto dal menu e modificarne il nome e/o il prezzo tramite le classi corrispondenti.
  {*AggiungiPiattoGUI, RimuoviPiattoGUI, ModificaNomePiattoGUI, ModificaPrezzoPiattoGUI*}
- **CamerieregUI**: Sezione dedicata al Cameriere nella quale è possibile prendere gli ordini dei clienti, modificarli e inviarli al cuoco una volta finalizzati. È inoltre possibile visualizzare la lista di piatti ordinati.
  {*PrendiOrdineGUI, ModificaOrdineGUI*}
- **CuocoGUI**: Sezione dedicata al Cuoco nella quale si ha accesso alla lista degli ordini finalizzati i quali vengono eliminati ed inviati al Responsabile di cassa una volta evasi.
  {*ListaPiattiGUI*}

- **ResponsabileDiCassaGUI**: Sezione dedicata al Responsabile di Cassa nella quale si possono visionare gli ordini completati e si possono stampare gli scontrini relativi agli ordini in un file .txt.
- **ErrorGUI**: Schermata che avverte l'utente in caso di errore durante l'utilizzo del programma.

2.2 Backend

- **Menu**: Classe che gestisce i piatti contenuti all'interno del menu e li memorizza all'interno di un file .txt.
  - **Attributi:**
    - *listaPiatti:*
      La lista dei piatti nel menu.
    - *fileNameMenu:*
      Il nome del menu salvato in formato .txt.
  - **Metodi:**
    - ***Menu***:
      Costruttore della classe che inizializza l'attributo listaPiatti con i patti presenti nel file .txt.
    - *getListaPiatti*:
      Getter per ricevere in output la lista dei piatti presenti attualmente nel menu.
    - *addPiattoDalMenu*:
      Metodo per aggiungere i piatti presenti nel file .txt, nell'ArrayList listaPiatti senza riaggiungerli al file di testo.
    - *addPiatto*:
      Metodo per aggiungere i piatti presenti nel file .txt, nell'ArrayList listaPiatti aggiungendoli anche nel file di testo.
    - *removePiatto*:
      Metodo utilizzato per rimuovere uno specifico piatto dal menu e dal file .txt.
    - *modificaNome*:
      Metodo per modificare il nome di uno specifico piatto nell' ArrayList listaPiatti e nel menu .txt.

    * *modificaPrezzo*:
      Metodo per modificare il nome di uno specifico piatto nell' ArrayList listaPiatti e nel menu .txt.

    * *formattaMenu*:
      Metodo che formatta la stringa da scrivere all'interno del file .txt.

    * *creaMenuCartaceo*:
      Metodo che crea il file .txt utilizzando le stringhe formattate dal metodo formattaMenu.

- **Piatto:** Classe che contiene le informazioni sul singolo piatto.
  - **Attributi:**
    * *nome*:
      Nome del piatto.

    * *prezzo*:
      Prezzo del piatto.

  - **Metodi:**
    * ***Piatto***:
      Costruttore della classe che inizializza i due attributi con i valori forniti.

    * *getNome*:
      Getter per ricevere in output il nome del piatto.

    * *setNome*:
      Setter per modificare il nome del piatto.

    * *getPrezzo*:
      Getter per ricevere in output il prezzo del piatto.

    * *setPrezzo*:
      Setter per modificare il prezzo del piatto.

    * *toString*

- **Ordine:** Classe che contiene le informazioni relative al singolo ordine.
  - **Attributi:**
    - *num*:
      Numero identificativo dell'ordine.
    - *paper*:
      Istanza della classe Scontrino.
    - *tot*:
      Attributo utilizzato nella creazione dello scontrino, per il prezzo totale dell'ordine.
    - *prezzoTot*:
      Attributo utilizzato per il prezzo totale dell'ordine.
    - *piattiOrdinati*:
      Attributo utilizzato per la lista dei piatti collegati all'ordine.
  - **Metodi:**
    - ***Ordine***:
      Costruttore della classe che inizializza l'istanza "paper", il numero identificativo + 1 (per distinguerlo dal precedente ordine), il prezzo totale a 0 e un nuovo ArrayList dove conserveremo i piatti ordinati.
    - *addPiattoOrdinato*:
      Metodo che aggiunge il piatto ordinato all'ArrayList piattiOrdinati e incrementa prezzoTot con il valore dell'attributo prezzo del piatto.
    - *removePiattoOrdinato*:
      Metodo che rimuove il piatto ordinato dall'ArrayList piattiOrdinati e decrementa l'attributo prezzoTot con il valore dell'attributo prezzo del piatto.
    - *addPiattoScontrino*:
      Metodo utilizzato per aggiungere al file .txt dello scontrino le stringhe relative al nome e al prezzo del piatto da inserire.
    - *getPrezzoTot*:
      Getter per ricevere in output il prezzo totale dell'ordine.

        * *getNum*:
          Getter per ricevere in output il numero identificativo dell'ordine.

        * *setNum*:
          Setter per modificare il numero identificativo dell'ordine.

        * *getPiattiOrdinati*:
          Getter per ricevere in output la lista dei piatti ordinati.

        * *clear*:
          Metodo utilizzato per eliminare ogni piatto dall'ordine.

        * *toString*

- **Scontrino:** Classe che gestisce la creazione e la stampa dello scontrino.
  - **Attributi:**
    * *scontrino*:
      Stringa che verrà trascritta sul file .txt dello scontrino.
  - **Metodi:**
    * ***Scontrino***:
      Costruttore della classe che inizializza "scontrino" ad una stringa vuota.

    * *aggiungiRiga*:
      Metodo che aggiunge a "scontrino" la stringa formattata relativa ad un piatto.

    * *formattaRiga*:
      Metodo che formatta la stringa da aggiungere a "scontrino".

    * *stampa*:
      Metodo che stampa lo scontrino sotto forma di file di testo.

- **Chef:** Classe che gestisce il menu.
  - **Attributi:**
    * *mMenu*:
      Istanza della classe Menu.

- **Metodi:**
  - ***Chef***:
    Costruttore della classe che inizializza l'attributo mMenu.
  - *getMenu*:
    Getter per ricevere in output l'attributo mMenu
  - *aggiungiPiatto*:
    Metodo per aggiungere un piatto al menu (utilizzando il metodo della classe Menu).
  - *eliminaPiatto*:
    Metodo per eliminare un piatto dal menu (utilizzando il metodo della classe Menu).
  - *modificaNome*:
    Metodo per modificare il nome di un piatto dal menu (utilizzando il metodo della classe Menu).
  - *modificaPrezzo*:
    Metodo per modificare il prezzo di un piatto dal menu (utilizzando il metodo della classe Menu).

- **Cameriere:** Classe che gestisce i singoli ordini.
  - **Attributi:**
    - *ordine*:
      Istanza della classe Ordine.
  - **Metodi:**
    - ***Cameriere***:
      Costruttore della classe che inizializza l'attributo ordine.
    - *aggiungiNellOrdine*:
      Metodo per aggiungere un piatto nell'ordine (utilizzando il metodo della classe Ordine.
    - *rimuoviNellOrdine*:
      Metodo per rimuovere un piatto dall'ordine (utilizzando il metodo della classe Ordine.
    - *finalizzaOrdine*:
      Metodo che svuota l'intero ordine una volta completato.

        * *annullaOrdine*:
          Metodo che svuota l'intero ordine.

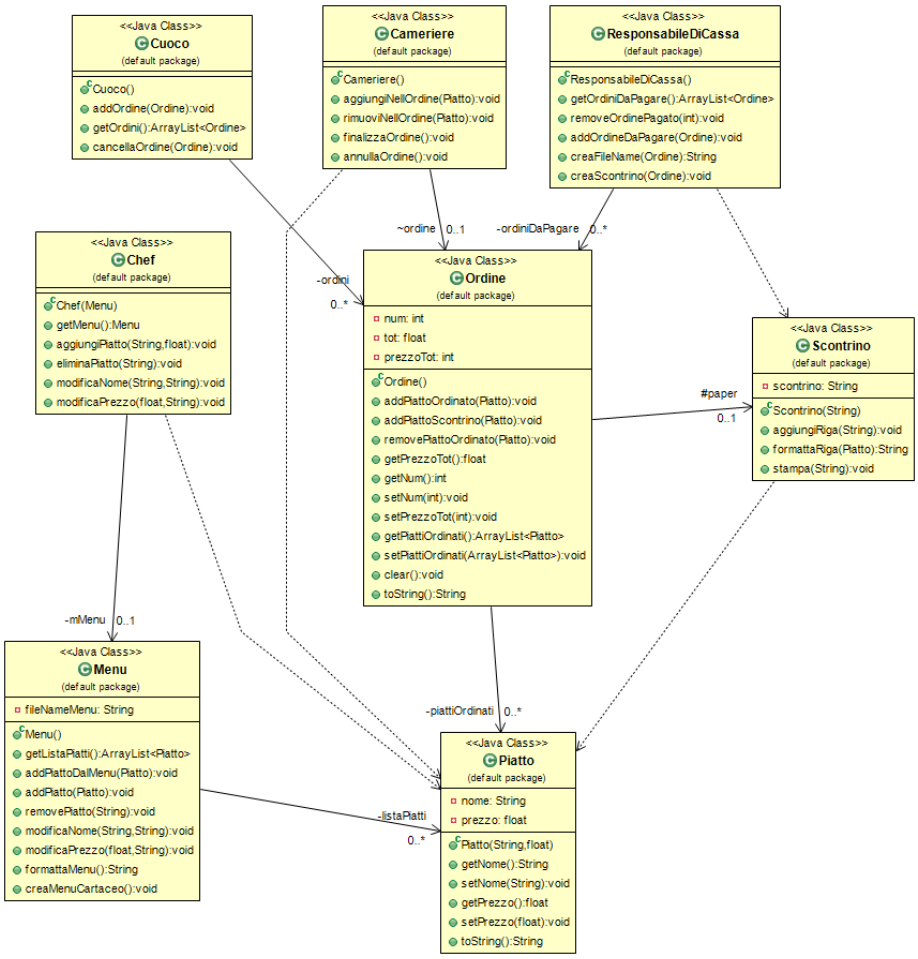
- **Cuoco:** Classe che gestisce gli ordini da evadere.
  - **Attributi:**
    * *ordini*:
      ArrayList contenente tutti gli ordini finalizzati.
  - **Metodi:**
    * ***Cuoco***:
      Costruttore della classe che inizializza l'ArrayList "ordini".
    * *addOrdine*:
      Metodo che aggiunge un'istanza della classe Ordine all'ArrayList "ordini".
    * *getOrdini*:
      Getter per ricevere in output tutti gli ordini contenuto nell'ArrayList.
    * *cancellaOrdine*:
      Metodo che rimuove un ordine dall'attributo "ordini".

- **ResponsabileDiCassa:** Classe che gestisce gli ordini evasi.
  - **Attributi:**
    * *ordiniDaPagare*:
      ArrayList contenente tutti gli ordini da pagare.
  - **Metodi:**
    * ***ResponsabileDiCassa***:
      Costruttore della classe che inizializza l'ArrayList "ordiniDaPagare".
    * *getOrdiniDaPagare*:
      Getter per ricevere in output l'attributo "ordiniDaPagare".
    * *removeOrdinePagato*:
      Metodo per rimuovere un ordine una volta pagato.

* *addOrdineDaPagare*:
  Metodo per aggiungere all'ArrayList un ordine da pagare.

* *creaFileName*:
  Metodo per creare il nome del file di testo dello scontrino.

* *creaScontrino*:
  Metodo che crea il file .txt dello scontrino ed elimina dall'Arraylist l'ordine.



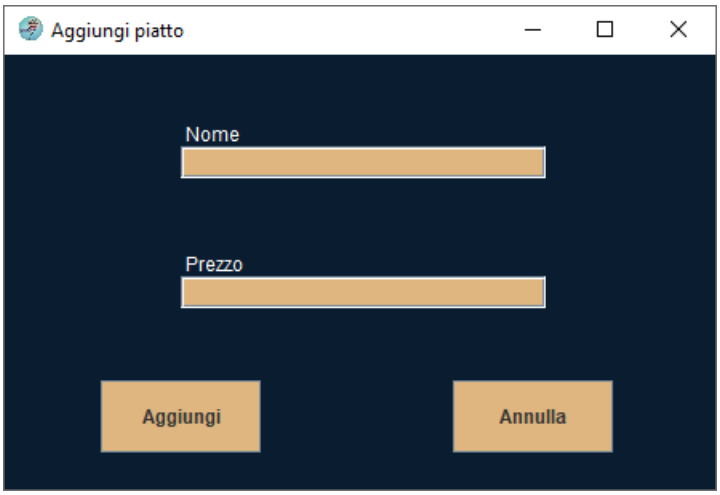
3 Descrizione delle funzionalità

3.1 Visualizzazione del menù iniziale

Il menu iniziale viene mostrato tramite un frame della classe JFrame all'avvio del programma, il quale è riempito con dei buttons della classe JButton che permettono di entrare nelle sezioni dei vari dipendenti, aggiornare il menu del ristorante, mostrato in primo piano nella finestra e uscire dal programma. Il menu del ristorante viene mostrato tramite un text pane della classe JTextPane il quale viene riempito all'avvio del programma con l'intero testo che si trova all'interno del file di testo "MenuCartaceo.txt".

3.2 Sezione Chef

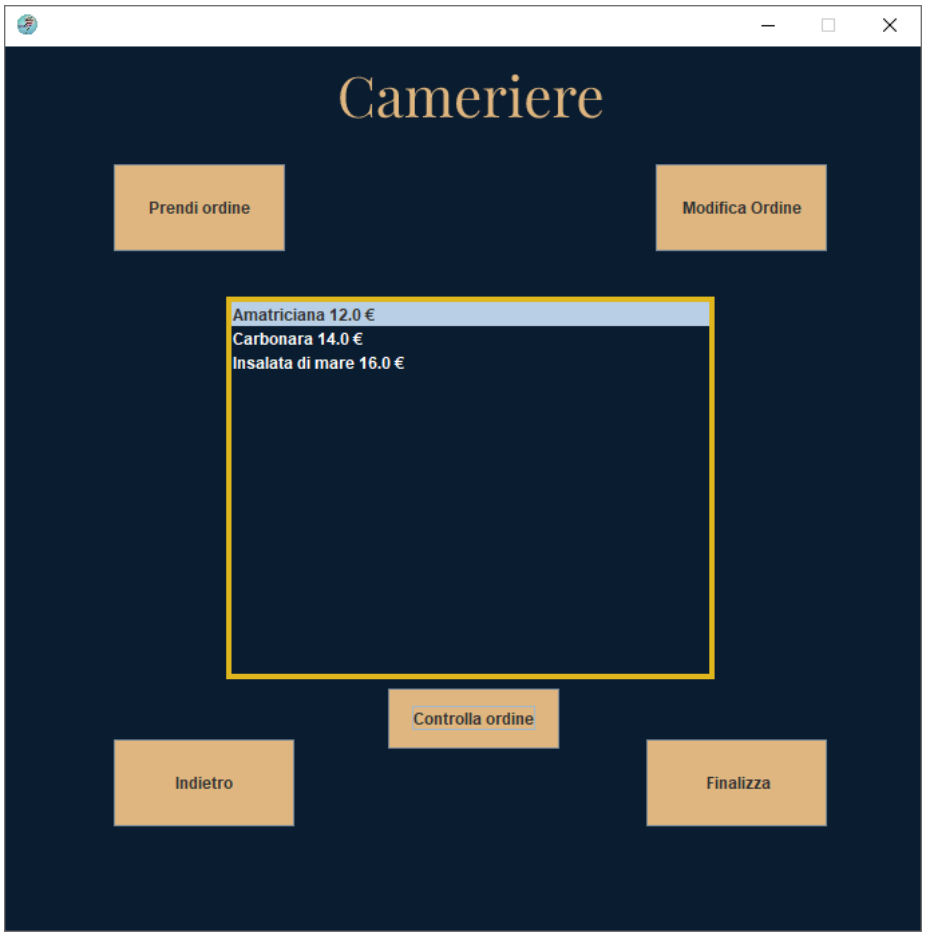
La sezione Chef gestisce il menu tramite un'istanza della classe stessa, attraverso la quale può intervenire sul file di testo "MenuCartaceo.txt" contenente i vari piatti che saranno poi disponibili all'interno del programma.

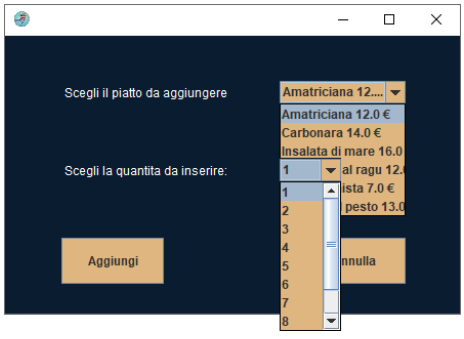


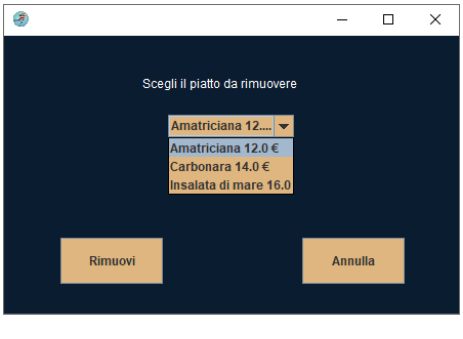
Le finestre dei restanti tre metodi sono analoghe

3.3 Sezione Cameriere

La sezione Cameriere gestisce la creazione di ordini tramite un'istanza della classe stessa. In un ordine si possono aggiungere o rimuovere piatti e al termine del processo di modifica si può finalizzare l'ordine per mandarlo al Cuoco. Durante il processo si Prima di finalizzarlo il cameriere può mostrare la lista delle pietanze ordinate con il tasto "Controlla ordine".

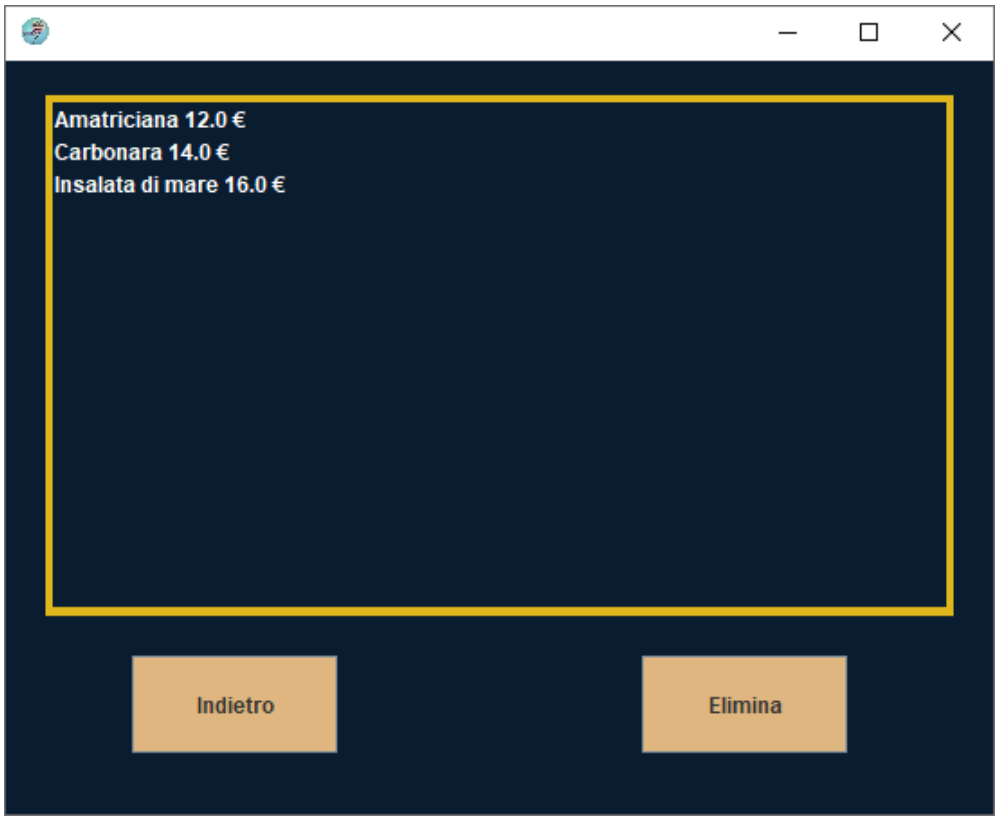

Cameriere
Prendi ordine
Modifica Ordine
Amatriciana 12.0 €
Carbonara 14.0 €
Insalata di mare 16.0 €
Controlla ordine
Indietro
Finalizza


Scegli il piatto da aggiungere
Amatriciana 12....
Amatriciana 12.0 €
Carbonara 14.0 €
Insalata di mare 16.0
Scegli la quantita da inserire:
1
2
3
4
5
6
7
8
Aggiungi


Scegli il piatto da rimuovere
Amatriciana 12....
Amatriciana 12.0 €
Carbonara 14.0 €
Insalata di mare 16.0
Rimuovi
Annulla

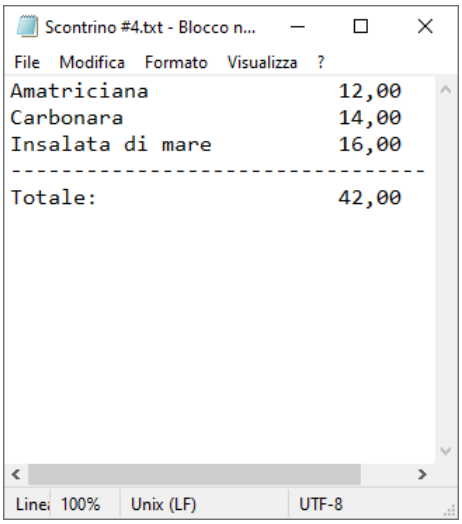
3.4 Sezione Cuoco

La sezione Cuoco gestisce gli ordini inviati da Cameriere rimuovendo i piatti all'interno dell'ordine selezionato uno per volta. All'interno dell'interfaccia grafica si possono selezionare i vari ordini e visualizzare i piatti da cucinare. Quando tutti i piatti in un ordine vengono evasi, l'ordine verrà mandato al responsabile di cassa.


Amatriciana 12.0 €
Carbonara 14.0 €
Insalata di mare 16.0 €
Indietro
Elimina

3.5 Sezione Responsabile di cassa

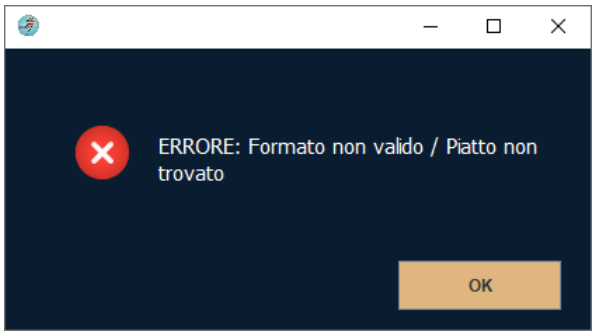
La sezione Responsabile di cassa gestisce gli ordini completati dal cuoco, permettendo la creazione dello scontrino. Dopo aver premuto il pulsante genera scontrino, questo verrà salvato in un file .txt nella directory principale del programma.



Il file .txt generato dal Responsabile di Cassa

3.6 Schermate di Errore

Durante la normale esecuzione del programma si possono creare vari errori, i quali vengono presentati all'utente tramite schermate di errore.



Le varie schermate di errore sono simili tra loro

4 Referenti di sviluppo

Il programma è stato interamente sviluppato da Samuele Bella e Francesco De Persio in cooperazione.