MANUALE TECNICO

BANCORA Davide	743662	Como
CASALNOVO Giacomo	740003	Como
DONATO Benedetta	742957	Como
DUBINI Emanuele	740954	Como

Indice

Benvenuto in “Centro Vaccinale”

EatAdvisor **è un software che offre un sistema di valutazione dei ristoranti basato sui giudizi espressi dai clienti. E’ molto potente e facile da usare.**

Presentazione tecnica

EatAdvisor è il miglior programma di inserimento e ricerca di giudizi di ristoranti ed è composto da due potenti e incredibili applicazioni: *Ristoratori e Clienti* che gestiscono in maniera specifica le due tipologie di utenti.

- dei formati dei file
- delle scelte algoritmiche effettuate

Segue l’appendice A che illustra lo schema delle classi in Java utilizzate.

Struttura del progetto

L’applicazione *Ristoratori* inserisce tutte le informazioni relative ai ristoranti che vengono registrati all’interno di un’opportuna struttura dati che viene memorizzata in un file denominato `EatAdvisor.dati`.
L’applicazione *Clienti* accede al file `EatAdvisor.dati` in lettura per poter effettuare le operazioni di ricerca, selezione, visualizzazione e in scrittura per effettuare l’operazione di attribuzione di un giudizio.
L’applicazione *Clienti*, inoltre, offre la possibilità agli utenti di registrarsi e inserisce le informariendly” il programma, come la visualizzazione della data.
Ne è stato dato dettaglio nella documentazione delle classi.
.

Documentazione *ServerCV*

Questa classe rappresenta la prima parte di EatAdvisor, quella in cui il file `EatAdvisor.dati` viene riempito con i dati dei ristoranti.
Il metodo *main* (riportato sotto) permette l'esecuzione dell'interfaccia grafica:

Il metodo *main* usato per far partire l'applicazione contiene dei metodi di controllo contenuti nel *Package ProgUtili:* ***checkEatAdvisor()*** e ***checkUtenti()***.
Vengono controllati i due files dei dati e se non sono presenti vengono creati, in questo modo il programma non va ma i in eccezione

La classe ristoratori contiene inoltre tutti i *Listener* per il corretto utilizzo dell'interfaccia grafica e dei menu a tendina.

Interessante notare che Ristoratori è una classe che implementa l'interfaccia serialazible, in questo modo i dati dei ristoranti si possono serializzare in un ArrayList per essere richiamati in modo agevole.

Documentazione *Common*

In questo *package* sono state create le classi che permettono l'utilizzo di classi e metodi comuni a entrambe le applicazioni ed alcuni metodi utili nel programma.

Classe *ProgUtili*

In questa classe sono presenti alcuni metodi utili come ad esempio quelli riportati qui:

Classe *Giudizio*

La classe Giudizio è stata usata per definire i giudizi da assegnare ai ristoranti.
Qui ovviamente viene generato l'oggetto Giudizio ed è stata ridefinito il metodo *toString* per poter stampare l'elenco dei giudizi includendo il nome del ristorante, il nickname del cliente che lo ha scritto e la doppia valutazione stelle/testo.

Classi *EatAdvisorException* e *UtentiException*

Sono due semplici classi di eccezioni quando qualcosa non funziona come dovrebbe.
Quando accade qualche errore e vengono sollevate, fanno uscire l'utente dal programma lanciando una semplice chiamata di sistema: *System.exit(-1)*.

Documentazione *ClientCV*

Qui c'è tutta la parte più corposa di EatAdvisor.

Il metodo *main* usato per far partire l'applicazione (come Ristoratori) contiene dei metodi di controllo contenuti nel *Package ProgUtili:* ***checkEatAdvisor()*** e ***checkUtenti()***. Vengono controllati i due files dei dati e se non sono presenti vengono creati, in questo modo il programma non va ma i in eccezione

Come la classe *Ristoratori*, anche *Clienti* contiene inoltre tutti i *Listener* per il corretto utilizzo dell'interfaccia grafica.

Inoltre la classe Clienti contiene alcune cose veramente interessanti. Innanzitutto Nella finestra di login, per accedere all'area riservata il campo password presenterà i caratteri nascosti, per sicurezza, usando la classe *JPasswordField*:

Sono stati inseriti i *tooltips* sui bottoni e sui cambi per aiutare l'utente negli inserimenti.

APPENDICE A

Classe *Ristoratori*

Classe *Common*

Classe *Clienti*