

JavaDoc - TheKnife Application

Package: theknife

Classe: TheKnife

Entry point dell'applicazione console TheKnife.

Gestisce l'interfaccia utente attraverso menu testuali e invoca le API del GestoreDati per le operazioni sui dati. I dati vengono salvati automaticamente alla chiusura tramite shutdown hook.

Metodi principali:

- `public static void main(String[] args)` - Entry point dell'applicazione
- `private static void menuIniziale()` - Mostra il menu principale con opzioni login/registrazione/guest
- `private static Utente login()` - Gestisce il processo di autenticazione
- `private static Utente registrazione()` - Gestisce la registrazione di nuovi utenti
- `private static void menuGuest()` - Menu per utenti non autenticati
- `private static void menuCliente(Utente u)` - Menu per clienti autenticati
- `private static void menuRistoratore(Utente u)` - Menu per ristoratori autenticati

Metodi di utilità:

- `private static int leggiInt(String messaggio)` - Input robusto per numeri interi
 - `private static double leggiDouble(String messaggio)` - Input robusto per numeri decimali
 - `private static boolean leggiBoolean(String messaggio)` - Input robusto per valori booleani
 - `private static String leggiNonVuota(String messaggio)` - Input robusto per stringhe non vuote
-

Classe: GestoreDati

Gestisce caricamento/salvataggio dei CSV e fornisce le API richieste dalla specifica.

Gestisce i file: ristoranti.csv, utenti.csv, recensioni.csv, preferiti.csv

Campi:

- `public final Map<Integer, Ristorante> ristoranti` - Mappa dei ristoranti indicizzati per ID
- `public final Map<String, Utente> utenti` - Mappa degli utenti indicizzati per username
- `public final Map<Integer, Recensione> recensioni` - Mappa delle recensioni indicizzate per ID
- `public final Map<String, Map<Integer,String>> preferiti` - Preferiti per utente con note

Metodi principali:

- `public void carica()` - Carica tutti i dati dai file CSV
- `public void salva()` - Salva tutti i dati sui file CSV
- `public static String sha256(String s)` - Calcola hash SHA-256 di una stringa

API di ricerca:

- `public List<Ristorante> cercaRistorante(String paeseOCitta, String tipoCucina, Integer prezzoMin, Integer prezzoMax, Boolean delivery, Boolean prenotazione, Double mediaMin)` - Ricerca ristoranti con filtri multipli
- `public String visualizzaRistorante(Ristorante r)` - Formatta i dettagli di un ristorante in tabella ASCII
- `public String visualizzaRecensioni(int ristoranteId)` - Formatta le recensioni di un ristorante

API utenti:

- `public Utente registrazione(String username, String password, String nome, String cognome, String dataN, String domicilio, Ruolo ruolo)` - Registra un nuovo utente
- `public Utente login(String username, String password)` - Autentica un utente

API preferiti (cliente):

- `public boolean aggiungiPreferito(String username, int ristoranteId)` - Aggiunge ristorante ai preferiti
- `public boolean rimuoviPreferito(String username, int ristoranteId)` - Rimuove ristorante dai preferiti
- `public List<Ristorante> visualizzaPreferiti(String username)` - Lista dei ristoranti preferiti
- `public boolean aggiornaNotaPreferito(String username, int ristoranteId, String nota)` - Aggiorna nota di un preferito

API recensioni (cliente):

- `public Recensione aggiungiRecensione(String username, int ristoranteId, int stelle, String testo)` - Aggiunge una recensione
- `public boolean modificaRecensione(String username, int recensioneId, int nuoveStelle, String nuovoTesto)` - Modifica una recensione esistente
- `public boolean eliminaRecensione(String username, int recensioneId)` - Elimina una recensione

API ristorante:

- `public Ristorante aggiungiRistorante(String proprietario, String nome, String nazione, String citta, String indirizzo, double lat, double lon, double prezzoMedio, boolean delivery, boolean prenotazione, String tipoCucina)` - Aggiunge un nuovo ristorante
- `public String visualizzaRiepilogo(String proprietario)` - Riepilogo recensioni per ristoranti del proprietario

- `public boolean rispostaRecensioni(String proprietario, int recensioneId, String risposta)` - Risponde a una recensione

Metodi di ordinamento:

- `public List<Ristorante> ordinaPerPrezzo(List<Ristorante> in)` - Ordina per prezzo crescente
 - `public List<Ristorante> ordinaPerNome(List<Ristorante> in)` - Ordina alfabeticamente per nome
 - `public List<Ristorante> ordinaPerMediaRecensioni(List<Ristorante> in)` - Ordina per media recensioni
-

Classe: Ristorante

Modello dati per ristorante.

Campi:

- `public int id` - Identificativo univoco
- `public String nome` - Nome del ristorante
- `public String nazione` - Nazione di ubicazione
- `public String citta` - Città di ubicazione
- `public String indirizzo` - Indirizzo completo
- `public double lat` - Latitudine
- `public double lon` - Longitudine
- `public double prezzoMedio` - Prezzo medio in euro
- `public boolean delivery` - Disponibilità servizio delivery
- `public boolean prenotazione` - Disponibilità prenotazione online
- `public String tipoCucina` - Tipo di cucina
- `public String proprietario` - Username del ristoratore proprietario

Costruttore:

- `public Ristorante(int id, String nome, String nazione, String citta, String indirizzo, double lat, double lon, double prezzoMedio, boolean delivery, boolean prenotazione, String tipoCucina, String proprietario)` - Costruttore completo
-

Classe: Utente

Dati anagrafici di un utente.

La password è memorizzata come hash SHA-256.

Campi:

- `public final String username` - Username (immutabile)
- `public String passwordHash` - Hash SHA-256 della password
- `public String nome` - Nome dell'utente
- `public String cognome` - Cognome dell'utente
- `public String dataNascita` - Data di nascita (formato yyyy-MM-dd)
- `public String domicilio` - Città di domicilio
- `public Ruolo ruolo` - Ruolo dell'utente (CLIENTE/RISTORATORE)

Costruttore:

- `public Utente(String username, String passwordHash, String nome, String cognome, String dataNascita, String domicilio, Ruolo ruolo)` - Costruttore completo
-

Classe: Recensione

Modello dati per recensione.

Campi:

- `public int id` - Identificativo univoco
- `public int ristoranteId` - ID del ristorante recensito
- `public String username` - Username dell'autore
- `public int stelle` - Valutazione da 1 a 5 stelle
- `public String testo` - Testo della recensione
- `public String risposta` - Risposta del ristorante (opzionale)

Costruttore:

- `public Recensione(int id, int ristoranteId, String username, int stelle, String testo, String risposta)` - Costruttore completo
-

Enum: Ruolo

Ruolo dell'utente nella piattaforma.

Valori:

- `CLIENTE` - Utente che può cercare ristoranti, lasciare recensioni e gestire preferiti
- `RISTORATORE` - Utente che può gestire ristoranti e rispondere alle recensioni

Classe: AsciiTable

Utility minimale per generare tabelle ASCII in console.

Non dipende da librerie esterne e fornisce formattazione tabellare per l'output console.

Metodi:

- `public static String render(List<String> headers, List<List<String>> rows)` - Genera una tabella ASCII formattata con intestazioni e righe

Metodi privati di supporto:

- `private static String line(List<Integer> widths)` - Genera linea separatrice
- `private static String row(List<Integer> widths, List<String> cells)` - Genera riga di dati
- `private static String pad(String s, int w)` - Applica padding a destra per allineamento