

Esercizio 1: Gioco del Tris (Tic-Tac-Toe)

Descrizione

Sviluppare un'applicazione Java Swing che implementi il gioco del Tris (Tic-Tac-Toe) con funzionalità di conteggio vittorie e reset partita.

Requisiti

1. Utilizzare JFrame come contenitore principale
2. Implementare una griglia 3x3 con pulsanti personalizzati
3. Alternare automaticamente i turni tra X e O
4. Verificare automaticamente la condizione di vittoria o pareggio
5. Visualizzare il punteggio di ciascun giocatore
6. Includere un pulsante per iniziare una nuova partita

Classi da implementare

- `TrisGame.java` : classe principale che estende JFrame
- `CellaTris.java` : classe personalizzata che estende JButton
- `AscoltaCella.java` : classe esterna che implementa ActionListener
- `MyLabel.java` : classe personalizzata per etichette del punteggio
- Altre classi ritenute necessarie

Funzionalità specifiche

1. Quando un giocatore clicca su una cella vuota, deve apparire il suo simbolo (X o O)
2. Le celle già occupate non devono essere modificabili
3. Dopo ogni mossa, verificare se il giocatore ha vinto o se la partita è in pareggio
4. Mostrare un messaggio di vittoria o pareggio e aggiornare il punteggio
5. Implementare almeno tre tipi di ascoltatori:
 - Classe anonima per il pulsante di reset
 - Classe interna per il cambio turno
 - Classe esterna per la gestione delle celle

Esercizio 2: Convertitore di Valute

Descrizione

Sviluppare un'applicazione Java Swing che permetta di convertire importi tra diverse valute con tassi di cambio personalizzabili.

Requisiti

1. Utilizzare JFrame come contenitore principale
2. Implementare campi di testo personalizzati per l'input e l'output
3. Permettere la selezione di almeno 5 valute diverse (Euro, Dollaro, Sterlina, Yen, Franco)
4. Visualizzare in tempo reale la conversione quando si modifica l'importo o la valuta
5. Includere la possibilità di modificare i tassi di cambio

Classi da implementare

- `ConvertitoreValute.java` : classe principale che estende JFrame
- `CampoValuta.java` : classe personalizzata che estende JTextField
- `AscoltaValuta.java` : classe esterna che implementa ActionListener
- `TassiCambio.java` : classe per gestire i tassi di cambio
- Altre classi ritenute necessarie

Funzionalità specifiche

1. La conversione deve avvenire in tempo reale durante la digitazione
2. Implementare un sistema di arrotondamento a due decimali
3. Permettere il salvataggio e il caricamento di tassi di cambio personalizzati
4. Impedire l'inserimento di caratteri non numerici nei campi di testo
5. Gestire gli ascoltatori di eventi utilizzando:
 - Classe anonima per i pulsanti di reset
 - Classe interna per la gestione degli eventi sui campi di testo
 - Classe esterna per la gestione dei tassi di cambio

Esercizio 3: Agenda Appuntamenti

Descrizione

Sviluppare un'applicazione Java Swing che permetta di gestire un'agenda di appuntamenti giornalieri con diverse funzionalità.

Requisiti

1. Utilizzare JFrame come contenitore principale
2. Implementare una visualizzazione degli appuntamenti per fascia oraria (dalle 8 alle 20)
3. Permettere l'aggiunta, la modifica e l'eliminazione di appuntamenti
4. Implementare un sistema di navigazione tra i giorni
5. Salvare gli appuntamenti in modo persistente

Classi da implementare

- `AgendaApp.java` : classe principale che estende `JFrame`
- `SlotAppuntamento.java` : classe personalizzata che estende `JPanel`
- `DialogoAppuntamento.java` : classe per il dialogo di creazione/modifica appuntamenti
- `AscoltaAgenda.java` : classe esterna che implementa `ActionListener`
- Altre classi ritenute necessarie

Funzionalità specifiche

1. Gli appuntamenti devono essere visualizzati in slot orari con colori diversi in base alla tipologia
2. Implementare un sistema di drag and drop per spostare gli appuntamenti
3. Permettere la ricerca di appuntamenti per keywords
4. Implementare una visualizzazione settimanale e mensile
5. Gestire gli ascoltatori di eventi utilizzando:
 - Classe anonima per i pulsanti di navigazione
 - Classe interna per la gestione degli slot orari
 - Classe esterna per la gestione dei dialoghi

Esercizio 4: Simulatore Bancomat

Descrizione

Sviluppare un'applicazione Java Swing che simuli il funzionamento di un ATM (bancomat) con diverse operazioni possibili.

Requisiti

1. Utilizzare `JFrame` come contenitore principale
2. Implementare un sistema di PIN per l'accesso (default: 1234)
3. Permettere operazioni di: prelievo, deposito, controllo saldo, cambio PIN
4. Visualizzare una ricevuta delle operazioni effettuate
5. Implementare limiti di prelievo e controllo di validità delle operazioni

Classi da implementare

- `SimulatoreBancomat.java` : classe principale che estende `JFrame`
- `DisplayBancomat.java` : classe personalizzata che estende `JTextField`
- `TastieraBancomat.java` : classe per gestire i pulsanti numerici
- `AscoltaOperazioni.java` : classe esterna che implementa `ActionListener`
- Altre classi ritenute necessarie

Funzionalità specifiche

1. La schermata iniziale deve richiedere l'inserimento del PIN
2. Dopo l'accesso, visualizzare un menu con le operazioni disponibili
3. Implementare controlli di validità (es. saldo insufficiente per prelievo)
4. Visualizzare messaggi di conferma o errore dopo ogni operazione
5. Gestire gli ascoltatori di eventi utilizzando:
 - Classe anonima per i pulsanti del menu principale
 - Classe interna per la tastiera numerica
 - Classe esterna per le operazioni bancarie