

Relazione di laboratorio

INFORMATICA

Classe:	4Di	Alunno:	BELLAMIA Antonio	Data:	19/mar/2024
Titolo dell'esercitazione:					
IMPLEMENTAZIONE DELLA GUI "CHIUSURA ANNO SCOLASTICO" IN JAVA					

Redigi la relazione sviluppando (almeno) i seguenti punti guida:

1. Analizza la Traccia e/o il Problema.

La traccia richiedeva l'implementazione di un JFrame da realizzare tramite le funzionalità della libreria 'swing', che contenesse 9 JTextField per l'inserimento dei voti (da 3 a 10) per ciascuna materia. Successivamente bisognava implementare un pulsante che dopo essere stato premuto, consentiva di popolare le JLabel Voto medio, facendo la media dei voti inseriti; Voto più basso; Voto più alto e Status (Il quale doveva mostrare "Promosso" se tutti i voti erano ≥ 6 , altrimenti non promosso).

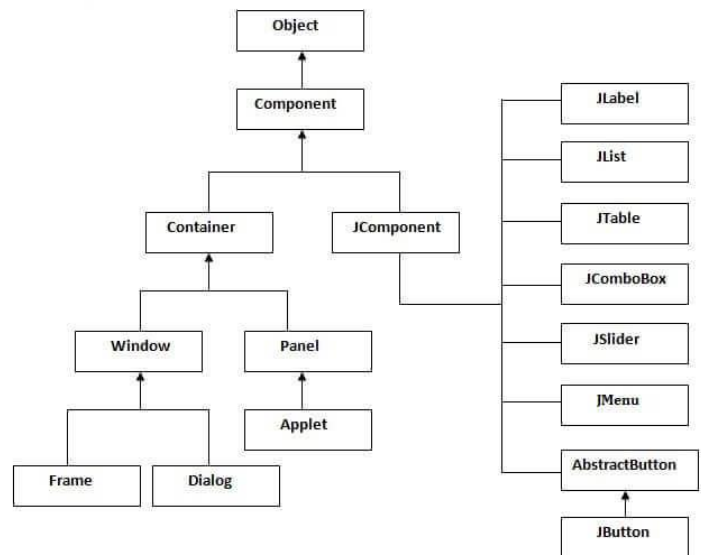
2. Inserisci i Cenni Teorici per rappresentare e/o risolvere il problema.

Per realizzare un'interfaccia grafica con il linguaggio Java ci dobbiamo servire di alcune classi specifiche importate attraverso le librerie Swing (javax.swing) e AWT (java.awt).

AWT è la libreria di interfaccia grafica originale fornita con il linguaggio di programmazione Java. Essa fornisce un insieme di classi per la creazione di finestre, bottoni, etichette, campi di testo e altri elementi di interfaccia utente. AWT utilizza i componenti di interfaccia utente nativi del sistema operativo sottostante, il che significa che le app AWT avranno un aspetto e un comportamento simile all'ambiente operativo in cui vengono eseguite. Swing, invece, è una libreria di componenti di interfaccia grafica più recente e più flessibile rispetto ad AWT. Essa è stata progettata per essere indipendente dalla piattaforma, il che significa che le app Swing avranno lo stesso aspetto e comportamento su qualsiasi piattaforma Java. Swing fornisce una vasta gamma di componenti di interfaccia utente, tra cui finestre, bottoni, pannelli, barre di scorrimento, caselle di controllo, menu e molti altri. Inoltre, Swing offre un'elevata personalizzazione e flessibilità nell'aspetto degli elementi di interfaccia utente.

Ogni componente grafico è di fatto un oggetto, creato a partire da una specifica classe. Tutte le classi che vanno a definire i componenti dell'interfaccia grafica ereditano dalla classe padre "**Component**", i metodi comuni forniti da questa classe includono la gestione degli eventi di input, la gestione della grafica e la gestione del layout. **Container** è una sottoclasse di Component che rappresenta un contenitore per altri componenti. Questa classe fornisce metodi per aggiungere, rimuovere e disporre

i componenti al suo interno. I contenitori possono essere organizzati attraverso i layout manager che determinano come i componenti sono disposti all'interno del contenitore. La classe container viene estesa da Window e da Panel. Le **window** rappresentano una finestra di livello superiore dell'applicazione. Le finestre sono contenitori specializzati che possono contenere altri componenti di interfaccia utente. Dalla classe window ereditano JFrame e le finestre di dialogo. **JFrame** è di fatto la finestra che contiene l'intera GUI. **JComponent**, invece, è una sottoclasse di Container e rappresenta la classe base per tutti i componenti Swing. Esistono varie classi derivate dalla più generica JComponent, alcune di esse sono:



-JButton: Rappresenta un pulsante cliccabile.

-JLabel: Rappresenta un'etichetta di testo o un'immagine non interattiva.

-JTextField: Fornisce un campo di testo modificabile per l'input dell'utente.

-JTextArea: Fornisce un'area di testo multi-linea modificabile.

-JComboBox: Fornisce una casella a discesa di selezione con opzioni predefinite.

-JCheckBox e JRadioButton: Rappresentano caselle di controllo e bottoni di opzione rispettivamente.

La classe JFrame mette a disposizione diversi metodi come ad esempio il metodo setTitle(String title) con il quale è possibile attribuire un titolo alla finestra. Il metodo setSize(int width, int height) serve a definire la dimensione di una finestra Swing.

Con il metodo setDefaultCloseOperation(int operation), è possibile configurare il comportamento predefinito della finestra quando l'utente tenta di chiuderla. Questo metodo offre varie opzioni, consentendo di definire il comportamento desiderato in risposta alla chiusura della finestra. Ad esempio, utilizzando setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE), l'applicazione terminerà automaticamente quando l'utente chiude la finestra principale.

Il metodo setVisible(boolean visible) serve a controllare se una finestra deve essere resa visibile o nascosta all'utente. Ad esempio, utilizzando setVisible(true), è possibile rendere visibile una finestra appena creata, consentendo agli utenti di interagire con il suo contenuto.

Il metodo add(Component comp) offre la possibilità di arricchire dinamicamente una finestra con nuovi componenti di interfaccia utente. Utilizzando questo metodo, si possono aggiungere bottoni, etichette, campi di testo e altri elementi alla finestra per fornire funzionalità aggiuntive o presentare informazioni all'utente.

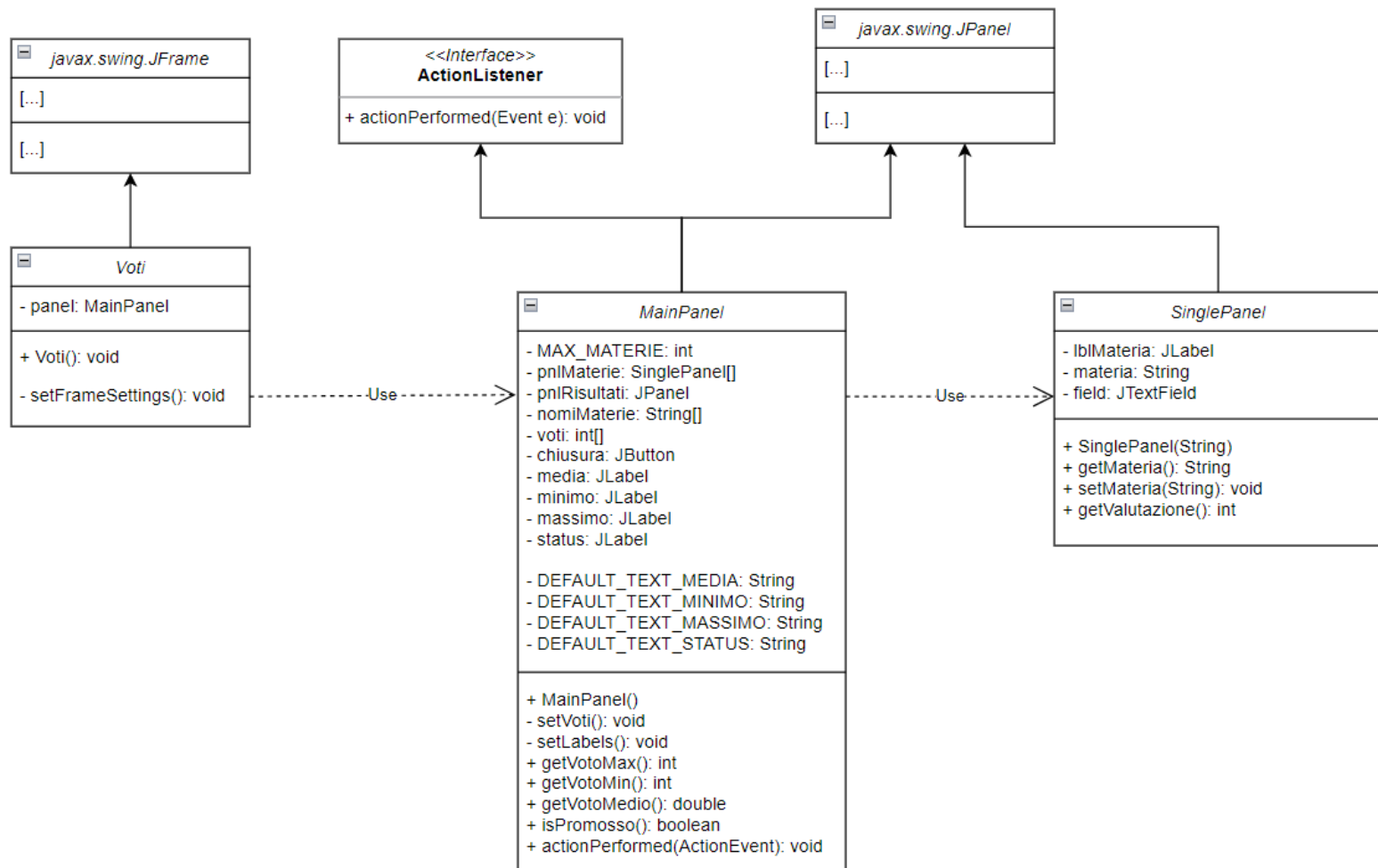
Infine, il metodo setLayout(LayoutManager manager) offre un controllo preciso sulla disposizione dei componenti all'interno di una finestra Swing. Attraverso l'uso di layout manager, gli sviluppatori possono organizzare i componenti in modo flessibile. Ad esempio, utilizzando un BorderLayout, è possibile assegnare a ciascun componente una posizione specifica all'interno della finestra. Il BoxLayout è un layout progettato per gestire il posizionamento dei componenti in una singola riga o colonna, mentre il GridLayout serve ad organizzare i componenti in una griglia regolare di righe e colonne. Ogni cella della griglia può contenere un singolo componente, e tutte le celle hanno le stesse dimensioni.

Le interazioni degli utenti con l'interfaccia grafica prendono il nome di eventi. Quando un evento viene generato, viene inviato a un oggetto chiamato "listener" che ha registrato interesse per quell'evento specifico. Il listener quindi esegue un'azione o una serie di azioni in risposta all'evento. Ad esempio, un listener di clic su un pulsante potrebbe eseguire un'azione quando l'utente preme il pulsante.

Gli eventi vengono gestiti da un listener, implementato con l'interfaccia ActionListener. ActionListener è un'interfaccia in Java utilizzata per gestire gli eventi generati dagli oggetti Swing. Questa interfaccia contiene un solo metodo, actionPerformed(ActionEvent e), che deve essere implementato da qualsiasi classe che desidera gestire gli eventi di azione. Quando un oggetto Swing genera un evento di azione (ad esempio, quando un pulsante viene premuto), viene chiamato il metodo actionPerformed() associato a quell'oggetto. All'interno di questo metodo, il codice per gestire l'azione desiderata può essere inserito.

3. Descrivi la Strategia Risolutiva che hai adottato.

DIAGRAMMA UML



Come è possibile notare nel diagramma UML, il programma si compone di tre classi: "Voti" che estende JFrame, "SinglePanel" e "MainPanel" che estendono JPanel ed in più, quest'ultimo, implementa l'interfaccia ActionListener. Il frame (rappresentato dalla classe "Voti"), si occupa semplicemente di istanziare un oggetto di tipo 'MainPanel' ed aggiungerlo al frame stesso; inoltre imposta le proprie dimensioni, il titolo del frame, l'opzione di uscita setDefaultCloseOperation(EXIT_ON_CLOSE) e viene settato come visibile.

MainPanel è il pannello generale aggiunto al frame, nel quale vengono inseriti tutti i componenti. Si è deciso, per una questione di organizzazione, di raggruppare ulteriormente i componenti in altri JPanel. Questi pannelli vengono aggiunti al MainPanel secondo un layout BorderLayout.Y_AXIS e quindi la gestione dei singoli componenti è demandata ai vari pannelli.

Vi sono 9 pannelli di tipo "SinglePanel" raggruppati in un'array, in più vi è un vettore di altrettanti elementi contenente i nomi delle 9 materie da combinare poi con i pannelli. Un vettore di interi invece serve ad immagazzinare i valori dei voti. In più c'è un altro pannello, posto sotto tutti gli altri, che contiene le label di media, minimo, massimo, status ed un pulsante "chiusura anno scolastico".

Il costruttore inizializza il pannello principale e tutti i suoi componenti. Aggiunge i singlePanel al pannello principale e inizializza i pulsanti e le label per la visualizzazione dei risultati. Il metodo setLabels() inizializza le etichette per visualizzare i risultati dei voti. I metodi getVotoMax(), getVotoMin(), getVotoMedio() ed isPromosso() calcolano rispettivamente il voto massimo, il voto minimo, il voto medio e se lo studente è promosso. Il metodo isPromosso ritorna il valore di 'falso' appena incontra un valore dell'array 'voti' inferiore al 6, altrimenti ritorna 'true'.

Il metodo setVoti() viene chiamato quando viene premuto il pulsante "Chiusura anno scolastico", tramite ActionListener. Esso recupera i voti inseriti dall'utente dai pannelli delle materie (richiamando il metodo getValutazione() di pnlMaterie[i]) e popola il vettore "voti" (che viene anche inizializzato qui). Se si verifica un'eccezione durante il recupero dei voti, viene visualizzato un messaggio di errore, tramite finestra di dialogo, contenente il testo relativo all'eccezione. I risultati vengono impostati su "n/d" (non disponibile) ed il vettore viene posto a null, l'elaborazione dei dati viene interrotta.

Il metodo actionPerformed(ActionEvent e) viene chiamato quando viene premuto il pulsante "Chiusura anno scolastico". Richiama setVoti() per recuperare i voti, quindi se non si sono verificati problemi (array!=null), modifica il testo delle varie label con il testo di default (attributi final) ed il risultato dei rispettivi metodi (massimo, minimo o media). Per quanto riguarda la label di status, se il valore ritornato da isPromosso è true, viene mostrato nella label il testo "promosso" e compare una finestra di dialogo con il messaggio "Complimenti, sei stato promosso!". In caso contrario, quando isPromosso ritorna il valore di false, viene mostrato nella label il testo "bocciato" e compare una finestra di dialogo con il messaggio "Purtroppo sei stato bocciato".

Tornando ai singlePanel, la scelta è stata dettata dalla semplicità richiesta nella gestione di questo tipo di pannello rispetto a dei pannelli puri. Tutti i pannelli dei voti sono istanze, oggetti, della stessa classe singlePanel, il che significa che se avviene un errore di codice in un singolo pannello, questo è comune a tutti i pannelli dello stesso tipo, rendendolo facilmente identificabile e risolvibile. Tutti i pannelli seguono lo stesso modello, perciò non possono esserci ambiguità nella gestione dei pannelli nel complesso. Ogni SinglePanel contiene un JLabel contenente il nome della materia ed un JTextField per l'inserimento del voto. Questi due componenti sono organizzati da un GridLayout composto da una riga e due colonne. Al costruttore viene passata come parametro una stringa, con il nome della materia da inserire nella label. Nel costruttore vengono istanziati ed aggiunti i componenti. Vi è il metodo setMateria che imposta, dopo gli opportuni controlli, il testo della label e vi è il corrispettivo getter.

Inoltre è presente il metodo `getValutazione`. Innanzitutto viene estratto il nome della materia eliminando il carattere ':' che caratterizza il testo della label. Viene inizializzata la variabile intera `v` che andrà a contenere il voto estratto dal field. In un try-catch viene ricavata la stringa dal `textField` e viene trasformata in intero con il metodo statico `parseInt` della classe `wrapped Integer`. Se la conversione in intero non va a buon fine, viene sollevata un'eccezione `NumberFormatException` con il testo "Non è possibile elaborare il valore inserito alla voce "+nome materia. Un ulteriore controllo verifica che il voto sia compreso tra 3 e 10, altrimenti solleva un'eccezione `IllegalArgumentException` con il testo "Il valore inserito alla voce "+nome materia+" deve essere compreso tra 3 e 10.". In questo modo tutti i controlli sono a carico del pannello singolo, andando a semplificare la struttura del codice del pannello principale. Il metodo in questione poi restituisce il voto.

4. Elenca gli Strumenti e/o il Materiale che hai utilizzato.

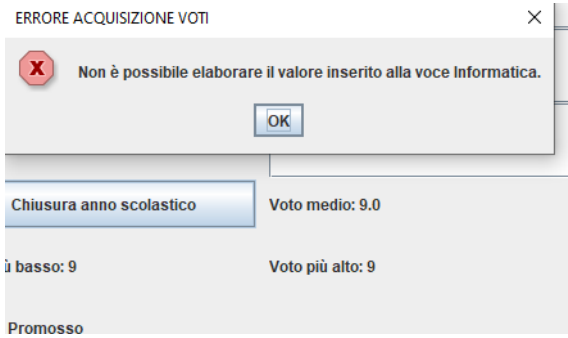
Per effettuare la nostra esercitazione ci siamo serviti dei computer messi a disposizione dalla scuola nel laboratorio 36. Sulle nostre macchine (o compilatori) abbiamo usufruito della Java Virtual Machine (JVM, messa a disposizione da Oracle, la società proprietaria del linguaggio Java) e del Java Development Kit (JDK, sempre fornito da Oracle). L'IDE (Integrated Development Environment) che abbiamo utilizzato è IntelliJ IDEA, nella sua versione community, sviluppata da JetBrains.

5. Descrivi le eventuali difficoltà che hai riscontrato e, se ci sei riuscito, come le hai superate.

Grazie alle lezioni teoriche ed alle esercitazioni in laboratorio, sono riuscito a risolvere il compito in autonomia e rapidamente.

6. Descrivi in che modo la Soluzione che hai implementato si potrebbe migliorare.

Per migliorare la fluidità dell'esperienza dell'utente, in presenza di un errore, sarebbe opportuno nel metodo `setVoti` della classe `MainPanel` invertire l'ordine in cui viene mostrata la finestra di dialogo con il reset delle label. Per come è strutturato attualmente il codice, il reset avviene solo dopo che l'utente ha liquidato la finestra di dialogo, lasciando, per questo periodo di tempo, sul frame principale i risultati delle operazioni precedenti. Sarebbe più piacevole, invece, avere dei valori resettati anche mentre la finestra di dialogo è mostrata.



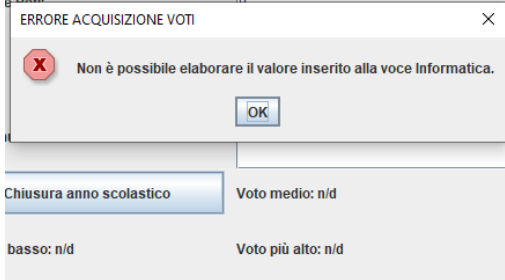
```

voti[i] = pnlMaterie[i].getValutazione();
} catch (Exception exc) {
    JOptionPane.showMessageDialog( parentComponent: null,

        media.setText(DEFAULT_TEXT_MEDIA + "n/d");
        minimo.setText(DEFAULT_TEXT_MINIMO + "n/d");
        massimo.setText(DEFAULT_TEXT_MASSIMO + "n/d");
        status.setText(DEFAULT_TEXT_STATUS + "n/d");

        voti=null;
        break;
    
```

PRIMA



```

media.setText(DEFAULT_TEXT_MEDIA + "n/d");
minimo.setText(DEFAULT_TEXT_MINIMO + "n/d");
massimo.setText(DEFAULT_TEXT_MASSIMO + "n/d");
status.setText(DEFAULT_TEXT_STATUS + "n/d");

JOptionPane.showMessageDialog( parentComponent: null, exc.getMessage(), title:

voti=null;
break;
    
```

DOPO

7. Descrivi in cosa senti di essere migliorato grazie a quest'esercitazione.

Come per ogni prova laboratoriale, posso dire di aver allenato le mie capacità di problem solving e di aver incrementato la mia familiarità con il linguaggio Java.

Bellamia Antonio
4D informatica