

Analisi esercizio “Ese120-GlobalSports”

CREARE UN’APPLICAZIONE CON UN’INTERFACCIA GRAFICA PER LA VENDITA DI UN PARTICOLARE PRODOTTO (TELEFONI, SCARPE, VIDEOGIOCHI...). DEVE ESSERE POSSIBILE VISIONARE LE FOTO DEI PRODOTTI E SCEGLIERLI; VISUALIZZARE IL COSTO TOTALE DEI PRODOTTI ACQUISTATI. UTILIZZARE ALMENO UNA FINESTRA DI DIALOGO.

Analisi del Problema (autore: Mattia Morelli-Analista)

Il problema chiede di realizzare un’applicazione per la vendita di prodotti; ho quindi scelto di specializzare la richiesta simulando un negozio sportivo denominato “**Global Sports**” col seguente slogan: “*Sport makes the world go round*”.

Gli obiettivi da realizzare possono essere sintetizzati in creare un’interfaccia grafica, gestire l’acquisto dei prodotti e calcolare la spesa totale.

L’applicativo dovrà contenere delle aree dove sarà possibile selezionare gli articoli (suddivisi per categoria) insieme ad un’ulteriore interfaccia contenente il riepilogo e la possibilità di modificare quanto scelto. Per ogni singolo articolo si potranno ordinare da 1 a 10 elementi, in modo da semplificare lo sviluppo. Inoltre non si potrà modificare la quantità selezionata, se non si elimina l’ordinazione e se ne rigenera un’altra. Una volta selezionato il tasto “*Acquista*” contenuto nel carrello, comparirà una finestra di conferma (*JOOptionPane*) che permetterà all’utente di finalizzare l’acquisto. La finestra di informazione finale ringrazia l’utente per aver scelto l’azienda “**Global Sports**”.

Realizzazione dell’interfaccia grafica (autore: Nicola China-Grafico)

L’interfaccia grafica dovrà essere semplice ed intuitiva. Sarà provvista di un pannello di intestazione contenente il logo ed in nome dell’azienda. Conterrà anche una barra con una scheda per ogni categoria di articolo, insieme ad un’ultima scheda con il riepilogo del carrello. Ogni singolo articolo dovrà essere presentato in una griglia di tre colonne in formato quadrato contenente le informazioni base dell’apposito articolo, quali: *Immagine* del prodotto; *Nome* e *prezzo*.

Gli articoli riepilogati nel carrello conterranno, insieme alle informazioni essenziali, la quantità acquistata e un pulsante per eliminare la richiesta in modo tale che l’utente non debba essere costretto a ricominciare l’ordine da capo qualora sbagliasse a selezionare un articolo. Le ulteriori specifiche sono presenti nel file apposito indirizzato ad uno studio grafico.

Coordinazione del progetto (autore: Amedeo Di Gaetano-Coordiatore)

Per semplificare la realizzazione del progetto, ho deciso di utilizzare lo strumento di controllo versione **Git**, combinato con la piattaforma **Github**. In questo modo i programmatori potranno lavorare parallelamente, in modo autonomo, per poi riunificare i codici realizzati e testare l'applicazione terminata. Sono stato creato due rami aggiuntivi al principale (*master*). Inoltre, la libreria utilizzata è "*Swing*", fornita tra le funzioni base di Java. Per realizzare le schede verrà utilizzata la classe *JTabbedPane*; i pannelli verranno implementati tramite *JPanel* o *JScrollPane*, in base alla necessità. Altri componenti utilizzati sono *JFrame*, *JLabel* e *JButton*.

Implementazione (autore: Andrea Sette-programmatore)

L'implementazione è stata suddivisa in due parti tra i due programmatori, che hanno lavorato parallelamente per poter avere una maggiore chiarezza sul funzionamento delle diverse classi. Le classi assegnate al primo programmatore sono state:

- **PanelArticolo**
- **PanelGriglia**
- **PanelCarrello**

La prima classe è relativa alla visualizzazione di un prodotto composto dall'immagine e sotto ad essa l'apposito nome con il suo prezzo al di sotto.

La seconda classe permette l'aggiunta e la disposizione degli articoli all'interno della griglia precedentemente creata, prelevando le relative informazioni del prodotto da un file CSV.

La terza classe permette di avere una visione dei prodotti scelti per l'acquisto con le relative informazioni su prezzo e quantità. Se si è convinti di voler proseguire all'acquisto, si dovrà cliccare sull'apposito bottone "*Acquista*"

Implementazione (autore: Mattia Morelli-programmatore)

La seconda parte dell'implementazione consisteva nel trasformare le nostre idee in codice funzionante. Le classi assegnatemi sono:

- **PanelLogo**
- **FinestraPrincipale**
- **BarraSchede**

La prima classe, ovvero **PanelLogo** è riservata per la visualizzazione del logo e del nome dell'azienda che saranno inseriti nel pannello in alto all'interno della finestra del programma.

La classe **FinestraPrincipale** è la finestra che racchiude tutti i vari pannelli che sono presenti nell'interfaccia; nello specifico questa classe contiene il pannello per il logo, i pannelli per gli articoli e quello contenente il riepilogo.

L'ultima classe, ovvero **BarraSchede** è quella classe dove sono istanziati tutti i vari cataloghi disponibili con appositi immagine e attributi; cliccando su un catalogo si viene indirizzati in un pannello, dove verranno visualizzati tutti gli articoli acquistabili per quel catalogo.

Individuazione delle classi

Si è cercato di scomporre il programma in quante più classi possibili, in modo da poter lavorare in parallelo, semplificare la lettura del codice e le eventuali correzioni future.

Sono quindi state realizzate le seguenti classi:

- **ArticoloCarrello**: contiene l'articolo selezionato da visualizzare nel carrello;
- **BarraSchede**: contiene tutte le schede contenenti gli articoli divisi per categoria;
- **FinestraPrincipale**: contiene tutti gli elementi dell'interfaccia grafica;
- **Main**: contiene il metodo *main* e i vari *listener* (**ActionListener**, **MouseListener**);
- **PanelArticolo**: contiene gli elementi rappresentanti la visualizzazione di un determinato articolo;
- **PanelCarrello**: contiene il riepilogo degli articoli che si desidera acquistare;
- **PanelGriglia**: contiene gli articoli, disposti in una griglia;
- **PanelLogo**: contiene il logo ed il nome dell'azienda;

Inoltre, è stata realizzata un'altra classe, contenente utilità non specifiche per il problema:

- **CSVParser**: permette di leggere agevolmente un file .csv;

Individuazione di attributi e metodi di ogni classe

ArticoloCarrello (*attributi*):

protected PanelArticolo <i>articolo</i>	Articolo di riferimento
protected int <i>numero</i>	Quantità selezionata da acquistare
private final JLabel <i>img</i>	Immagine di anteprima
private final JLabel <i>info</i>	Informazioni dell'articolo
private final JLabel <i>quantita</i>	Quantità da acquistare
private final JButton <i>elimina</i>	Bottone per eliminare articolo

ArticoloCarrello (*metodi*): non ne possiede;

BarraSchede (attributi):

protected PanelCarrello <i>carrello</i>	Contiene il carrello
protected JScrollPane <i>scroll</i>	Contiene una categoria di articoli
private PanelGriglia <i>schede</i>	Pannello da inserire

BarraSchede (metodi): non ne possiede;**CSVParser (attributi):**

private final String[] <i>header</i>	Intestazione del file .csv
private final ArrayList<String[]> <i>rows</i>	Array contenente i campi delle righe

CSVParser (metodi):

public String <i>getElement(String element, int row)</i>	Restituisce la stringa di riga <i>row</i> e campo <i>element</i>
public static int <i>indexOf(String str, String[] list)</i>	Restituisce l'indice al quale si trova la stringa <i>str</i>
public String[] <i>getHeader()</i>	Restituisce l'intestazione del file .csv

FinestraPrincipale (attributi):

protected JPanel <i>pannello</i>	Andrà a costituire il <i>contentPane</i>
private final PanelLogo <i>logo</i>	Logo e nome azienda
protected BarraSchede <i>schede</i>	Pannello con schede articoli

FinestraPrincipale (metodi): non ne possiede;**Main (attributi):**

private FinestraPrincipale <i>gui</i>	Contiene l'interfaccia grafica del programma
--	--

Main (metodi):

public void <i>actionPerformed(ActionEvent e)</i>	Override da <i>ActionListener</i>
public void <i>mouseClicked(MouseEvent e)</i>	Override da <i>MouseListener</i>

PanelArticolo (attributi):

protected final String <i>nome</i>	Nome dell'articolo
protected final String <i>img</i>	Percorso dell'immagine
protected final float <i>prezzo</i>	Prezzo dell'articolo

PanelArticolo (metodi): non ne possiede;**PanelCarrello (attributi):**

private final JButton <i>bottona</i>	Pulsante acquista
---	-------------------

PanelCarrello (metodi): non ne possiede;**PanelGriglia (attributi):**

private PanelArticolo <i>articolo</i>	Articolo da aggiungere
--	------------------------

PanelGriglia (metodi): non ne possiede;

PanelLogo (attributi):

<code>private final JLabel logo</code>	Immagine del logo dell'azienda
<code>private final JLabel Nome_Azienda</code>	Immagine del nome dell'azienda

PanelLogo (metodi): non ne possiede;

Impostazione dei metodi costruttori

I metodi costruttori, elencati in ordine alfabetico, verranno descritti di seguito, divisi per blocchi operativi:

public **ArticoloCarrello**(*PanelArticolo prodotto, int quantita*):

- Vengono assegnati i valori dei parametri (*prodotto* e *quantita*) rispettivamente agli attributi *articolo* e *numero*;
- vengono istanziati gli altri attributi con i dovuti adattamenti, insieme alla definizione dell'**ActionCommand** e dell'**ActionListener**;
- vengono aggiunti i componenti al pannello.

public **BarraSchede**():

- Dopo aver creato una categoria di articoli, vengono impostate le impostazioni del pannello, il quale verrà poi aggiunto in una nuova scheda;
- la procedura viene ripetuta per ogni categoria da aggiungere, ad esclusione dell'aggiunta del carrello, la quale necessita di operazioni differenti.

public **CSVParser**(*String filename*):

- Viene aperto il file con un **BufferedReader**;
- viene prelevata l'intestazione;
- viene popolato l'**ArrayList** con tutti i campi, per ogni riga.

public **FinestraPrincipale**():

- Vengono istanziati gli attributi;
- dopo aver impostato il layout, vengono aggiunti il logo e la barra delle schede al pannello;
- quest'ultimo viene impostato come *contentPane* della finestra.

La classe **Main** non presenta costruttore esplicito.

public **PanelArticolo**(*String img, String nome, String prezzo*):

- Vengono istanziati gli attributi;
- vengono create tre **JLabel** (*name, price* e *image*);
- dopo aver impostato il layout, vengono aggiunte le label, insieme al **MouseListener**.

public **PanelCarrello()**:

- Viene impostato il layout;
- viene istanziato e impostato il bottone per l'acquisto;
- quest'ultimo viene infine aggiunto.

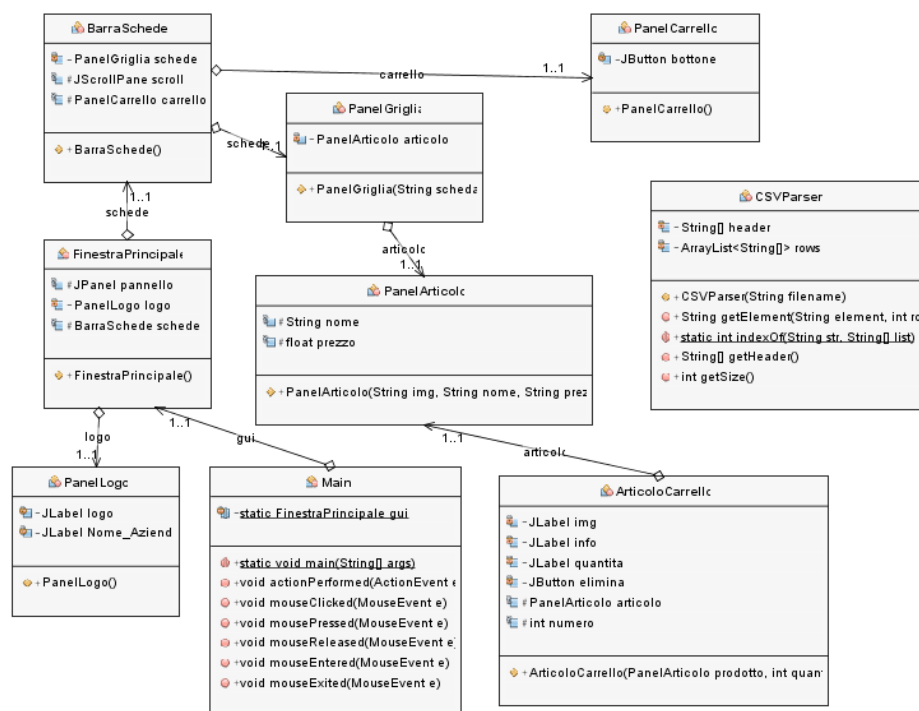
public **PanelGriglia(String sched**a):

- Viene impostato il layout;
- viene letto il file .csv della categoria di articoli;
- vengono creati e aggiunti gli articoli trovati.

public **PanelLogo()**:

- Vengono istanziati gli attributi;
- questi vengono infine aggiunti.

Rappresentazione grafica delle classi (in UML)



Il metodo main()

Il metodo *main* contiene solamente la creazione dell'istanza dell'interfaccia grafica, seguita da alcune impostazioni, come la dimensione della finestra e la dichiarazione dell'operazione di chiusura finestra.