

Classe Canvas

Il Canvas è una griglia di caratteri di forma rettangolare su cui possono agire diversi strumenti.

I campi di Canvas sono costituiti da due interi, corrispondenti al numero di righe (**R**) e colonne (**C**) del rettangolo, un Frame che rappresenta una singola griglia e fa riferimento all'attuale finestra di lavoro (**corrente**) e infine una pila di Frame che tiene traccia di tutte le eventuali modifiche apportate alla griglia (**registroFrame**). La pila è costruita attraverso la classe Stack presente nella libreria java.util.

Canvas contiene una classe privata Frame. Gli attributi del Frame sono due interi, **righe** e **colonne**, e un array di array di char, **caratteri**. La classe presenta due costruttori. Il primo costruttore riceve in ingresso due interi e inizializza **righe** e **colonne** rispettivamente. Inoltre inizializza caratteri ad un array di array di dimensioni righe x colonne contenente il carattere spazio. Il secondo costruttore riceve in ingresso un frame e ne costruisce un altro identico al frame in ingresso.

Il costruttore di Canvas riceve come parametri due interi, rispettivamente il numero di righe e di colonne della griglia. Quindi inizializza **R** e **C** con i parametri in ingresso, inizializza **registroFrame** attraverso il costruttore vuoto di Stack e gli aggiunge in *cima* un nuovo Frame di dimensioni R e C. Infine restituisce un puntatore al Frame che si trova in *cima* al **registroFrame** che viene memorizzato in **corrente**.

I metodi di Canvas:

- **public int getR()** : restituisce il numero di righe del canvas su cui viene applicato il metodo.
- **public int getC()** : restituisce il numero di colonne del canvas su cui viene applicato il metodo.
- **public char car(int x, int y)** : restituisce il carattere corrispondente alla cella (x,y) di questo canvas. Nel caso in cui x e y non rispettino le dimensioni del canvas, $0 < x < R$ e $0 < y < C$, viene visualizzato un opportuno messaggio di errore.
- **public boolean vuota(int x, int y)** : restituisce true se la cella (x,y) è vuota, cioè contiene il carattere spazio.
- **public boolean inBounds(int x, int y)** : restituisce true se e solo se i parametri x e y rispettano le dimensioni del canvas.
- **public void modifica(int x, int y, char c)** : sostituisce il carattere in posizione (x,y) con il carattere c in ingresso se e solo se i parametri x e y rispettano le dimensioni del canvas.
- **public void modifica(int x1, int y1, int x2, int y2, char c)** : traccia un segmento di estremi (x1,y1) e (x2,y2). Il metodo disegna solo la parte del segmento che rispetta le dimensioni del canvas. Per l'implementazione del metodo si confrontano la larghezza e l'altezza tra i due punti. Nel caso sia maggiore la prima le ascisse vengono incrementate di uno mentre l'incremento delle ordinate è riscalato secondo il coefficiente angolare della retta che passa per i due punti. Analogamente nel caso in cui sia maggiore l'altezza. Nel caso in cui i due punti coincidano il metodo si comporta come il metodo precedente per il punto in ingresso.
- **public void undo(int k)** : ripristina il canvas allo stato in cui si trovava prima delle ultime k modifiche. L'implementazione si basa sul metodo **pop()** della classe Stack che ad ogni chiamata rimuove il Frame che si trova in cima al **registroFrame**. Nel caso in cui **registroFrame** sia rimasto vuoto aggiungo un nuovo Frame di dimensioni pari a **R** e **C** del canvas su cui agisce il metodo. Infine **corrente** viene sovrascritto con il riferimento al Frame che si trova in cima alla pila.
- **public void addToHistory()** : questo metodo serve per memorizzare il Frame corrente prima che venga modificato. Il metodo agisce creando una copia del frame corrente,

aggiungendola al registro e infine aggiornando corrente con il riferimento al Frame che si trova in cima alla pila.

- **public Canvas copia()** : restituisce un nuovo canvas, copia di quello che chiama il metodo. Il registro del canvas restituito contiene solo il frame corrente e, sotto a questo, il frame vuoto.
- **public Canvas copia(int x1, int y1, int x2, int y2)** : Restituisce un nuovo canvas, copia del rettangolo di estremi (x1, y1), (x2,y2) del canvas che invoca il metodo. Il registro del canvas restituito contiene solo il frame corrente e, sotto questo, il frame vuoto.
- **public boolean equals (Object other)** : restituisce false se other non è istanza di Canvas. Nel caso in cui lo sia, chiama il metodo che segue per verificare l'uguaglianza tra i due oggetti di tipo Canvas.
- **public boolean equals(Canvas newCanvas)** : restituisce true se i frame correnti dei due Canvas sono uguali (stesso numero di righe, stesso numero di colonne e stessi caratteri nelle medesime posizioni).
- **public int hashCode()** : restituisce un intero pari alla somma dei caratteri del Frame corrente.
- **public String toString()** : restituisce una stringa corrispondente ai caratteri presenti nel Frame. Il carattere "a capo" serve per distinguere le varie righe del Frame.

Interfaccia Evento

Evento è un marker interface ed è implementata da due classi: EventoDiMouse ed EventoDiTastiera.

Classe EventoDiMouse

EventoDiMouse rappresenta il fatto che ci sia stata un'interazione tra il mouse e l'utente.

La classe presenta come attributi un booleano chiamato denominato dragging inizializzato a false che serve per controllare l'evento di DragEnd, infatti quest'ultimo si può verificare solo in seguito a un DragStart. Gli altri due attributi sono due campi interi, **x** e **y**, che indicano la posizione dell'evento all'interno del Canvas.

EventoDiMouse ha un unico costruttore che inizializza **x** e **y** con i due interi ricevuti in ingresso.

I metodi di EventoDiMouse sono:

- **public int posX()**: restituisce la coordinata x della posizione in cui si è verificato l'evento.
- **public int posY()**: restituisce la coordinata y della posizione in cui si è verificato l'evento.
- **public String toString()** : restituisce una stringa che esprime che si è verificato un evento di mouse in una determinata posizione.

Classe MouseMove

MouseMove rappresenta il fatto che il mouse è stato spostato e ora si trova in una data posizione.

MouseMove in quanto sottoclasse di EventoDiMouse, ha come unico costruttore quello ereditato dalla sua superclasse. Inoltre lascia invariato il campo dragging.

L'unico metodo che viene riscritto è **toString**, il quale non fa altro che specificare che l'evento di mouse in questione è di tipo `MouseMove`. Per il resto si comporta in modo analogo al metodo ereditato da `EventoDiMouse`.

Classe `MouseClicked`

`MouseClicked` rappresenta il fatto che l'utente ha premuto uno dei due tasti del mouse in una data posizione.

Oltre ai campi ereditati da `EventoDiMouse` presenta altri due campi di tipo boolean: **rightClick** e **doubleClick**. Questi due tasti rappresentano rispettivamente il fatto che è stato premuto il tasto destro e il fatto che ci sia stato un doppio click.

La classe ha unico costruttore che riceve in ingresso due interi che si riferiscono alla posizione e due boolean che si riferiscono agli eventi sopra descritti. Quello che fa il costruttore è inizializzare tutti gli attributi con i parametri ricevuti e ovviamente il **dragging** a false.

I metodi di `MouseClicked` sono:

- **public boolean rightClick()**: restituisce true se e solo se è stato premuto il tasto destro del mouse.
- **public boolean doubleClick()**: restituisce true se e solo se si è verificato un doppio click.
- **public String toString()**: restituisce un'opportuna stringa che descrive il fatto che ci sia stato un click col mouse e il tipo di click, semplice (non dice nulla) o doppio, e destro o sinistro.

Classe `DragStart`

`DragStart` corrisponde al fatto che l'utente ha premuto uno dei due tasti del mouse, e tenendolo premuto, sta spostando il mouse.

La classe è sottoclasse di `EventoDiMouse` e per questo eredita tutti i suoi attributi. Oltre a questi contiene un campo di tipo boolean, **right**. Quest'ultimo vale true se e solo se l'evento di drag è stato chiamato col tasto destro del mouse.

Il costruttore di `DragStart` riceve come parametri due interi (indicanti la posizione) e un boolean che si riferisce al tasto che è stato premuto (destro o sinistro). Il costruttore inizializza tutti i campi con i parametri in ingresso e cambia il valore di **dragging** a true.

I metodi aggiunti o riscritti in `DragStart` sono:

- **public boolean right()**: restituisce true se e solo se l'evento è stato chiamato col tasto destro del mouse.
- **public String toString()**: restituisce una stringa che descrive il fatto che c'è stato un evento di tipo `DragStart` e con quale tasto del mouse si è verificato.

Classe `DragEnd`

`DragEnd` rappresenta la fine di un evento di drag (momento in cui viene rilasciato il tasto del mouse).

Il costruttore riceve in ingresso i due interi che descrivono la posizione di fine drag. Inizializza quindi la posizione e ripristina il **dragging** a false.

L'unico metodo riscritto è:

- **public String toString()** : viene riscritto specificando che si tratta di un evento di tipo DragEnd.

Classe EventoDiTastiera

EventoDiTastiera rappresenta il fatto che l'utente abbia premuto un tasto sulla tastiera.

La classe presenta come unico attributo, **caratterePremuto**, di tipo char, che rappresenta il tasto della tastiera premuto dall'utente.

Il costruttore prende in ingresso un carattere e inizializza **caratterePremuto**.

I metodi di EventoDiTastiera sono:

- **public char car()**: restituisce il carattere premuto.
- **public String toString()**: restituisce una stringa che descrive il fatto che si è trattato di un evento di tastiera e il carattere digitato dall'utente.

Classe astratta Strumento

La classe Strumento reagisce ad alcuni eventi di mouse e come conseguenza modifica opportunamente il canvas.

Strumento ha un unico attributo **canvas** di tipo Canvas che sarà la base di lavoro per ogni strumento concreto definito come sottoclasse di Strumento.

Il costruttore riceve come parametro un canvas e inizializza di conseguenza **canvas**. Inoltre resetta sempre lo strumento allo stato iniziale grazie al metodo **reset()**, definito appunto in questa classe.

La classe risulta essere astratta dal momento che presenta due metodi astratti che verranno poi ridefiniti nelle varie sottoclassi.

I metodi di Strumento sono:

- **public abstract void ricevi(EventoDiMouse e)** : riceve un evento di tipo EventoDiMouse e modifica opportunamente il canvas su cui agisce strumento.
- **protected abstract void reset()** : resetta lo strumento allo stato iniziale.
- **public String toString()**: restituisce una stringa che descrive il tipo di strumento che sta lavorando sul canvas.

Strumento ha come sottoclassi: StrumentoDiDisegno, StrumentoDiSelezione e Salva.

Classe astratta StrumentoDiDisegno

Uno StrumentoDiDisegno è uno Strumento che serve per disegnare.

Questa classe presenta due attributi: **trattoDiDisegno** e **EVIDENZIATORE**. **trattoDiDisegno** è di tipo char e viene dichiarato static in modo tale da essere indipendente dall'oggetto stesso. Questo rappresenta il carattere utilizzato dallo strumento in questione per disegnare. **EVIDENZIATORE** è di tipo char e viene dichiarato anche lui static con l'aggiunta della clausola final, in modo tale che il suo valore non può essere mai modificato. **EVIDENZIATORE** viene utilizzato da alcuni strumenti di disegno per mostrare il punto del canvas rappresentate l'inizio del disegno.

StrumentoDiDisegno ha come unico costruttore quello ereditato da Strumento.

I metodi sono:

- **public static void setTratto(char c)**: modifica il valore di trattoDiDisegno con c.
- **public static char getTratto()** : restituisce il carattere che si sta utilizzando per il disegno.
- **protected abstract void reset()** : resetta lo strumento allo stato iniziale.
- **public String toString()** : restituisce una stringa che descrive lo strumento che sta lavorando sul canvas, compreso del tratto utilizzato durante il disegno.

StrumentoDiDisegno ha come sottoclassi: StrumentoDiDisegnoDrag, Poligonale, StrumentoDiDisegnoCentro e StrumentoDiDisegnoLibero.

Classe astratta StrumentoDiDisegnoDrag

StrumentoDiDisegnoDrag è uno StrumentoDiDisegno che risponde solo a eventi di DragStart e DragEnd; e solo nel caso in cui questi eventi si verifichino con il tasto sinistro del mouse.

E' una classe astratta che viene estesa da due classi concrete: Segmento e Riquadro.

Ha come attributi un DragStart di nome **pIniziale**, che è il primo EventoDiMouse da registrare, e un DragEnd di nome **pFinale**, che corrisponde al secondo EventoDiMouse da registrare.

Il suo costruttore viene ereditato direttamente da Strumento.

I metodi di questa classe sono:

- **protected abstract void disegna(DragStart pIniziale, DragEnd pFinale)** : disegna la figura per trascinamento dato un punto iniziale e finale.
- **public void ricevi(EventoDiMouse e)** : si occupa di rispondere agli input di mouse. Nel caso **e** si tratti di un DragStart inizializza pIniziale e, dopo il controllo che questo si sia verificato effettivamente col tasto sinistro, modifica il canvas evidenziando il punto di inizio. Prima di modificare il canvas salva il Frame nel registro. Nel caso, invece, **e** sia istanza di DragEnd e ci sia stato prima un evento di DragStart si può precedere con il disegno effettivo della figura attraverso **disegna()**. Alla fine vengono ripristinati DragStart a DragEnd a null.
- **protected void reset()** : ripristina DragStart e DragEnd a null e riporta il canvas allo stato iniziale.
- **public String toString()** : utilizza il toString della superclasse con l'aggiunta della stringa che si è trattato di uno strumento per trascinamento.

Classe segmento

Un segmento è uno StrumentoDiDisegnoDrag.

La classe è dichiarata di tipo final in modo tale che non può essere estesa da altre classi.

Il costruttore è ereditato da Strumento.

I metodi di Segmento sono:

- **protected void disegna(DragStart pIniziale, DragEnd pFinale)** : costruisce il segmento di estremi pIniziale e pFinale (corrispondenti ai due eventi di drag).

- **public String toString()** : restituisce una stringa che specifica che si è trattato dello strumento Segmento.

Riquadro

Riquadro è uno strumento concreto di StrumentoDiDisegno.

Il costruttore è ereditato da Strumento.

I metodi di Riquadro sono:

-