

Esercitazione 4

Java User Group Day

<http://www.jugtaas.org/jugday19/index.html>

A promotional poster for Java User Group Day. The background is a dark, moody landscape with a silhouette of a person running on the left. A bright sun or light source is in the center, creating a lens flare. At the top center is a circular logo featuring a steaming cup of coffee with a red cherry on top. The text is centered and reads: "18 maggio 2019", "Incontra la comunità di appassionati ed esperti del mondo Java™ Virtual Machine!", and "Vola al Java™ User Group Day del Trentino Alto Adige Südtirol ed incontra persone che si divertono, lavorano con Java™, Kotlin™ e Scala™".



18 maggio 2019

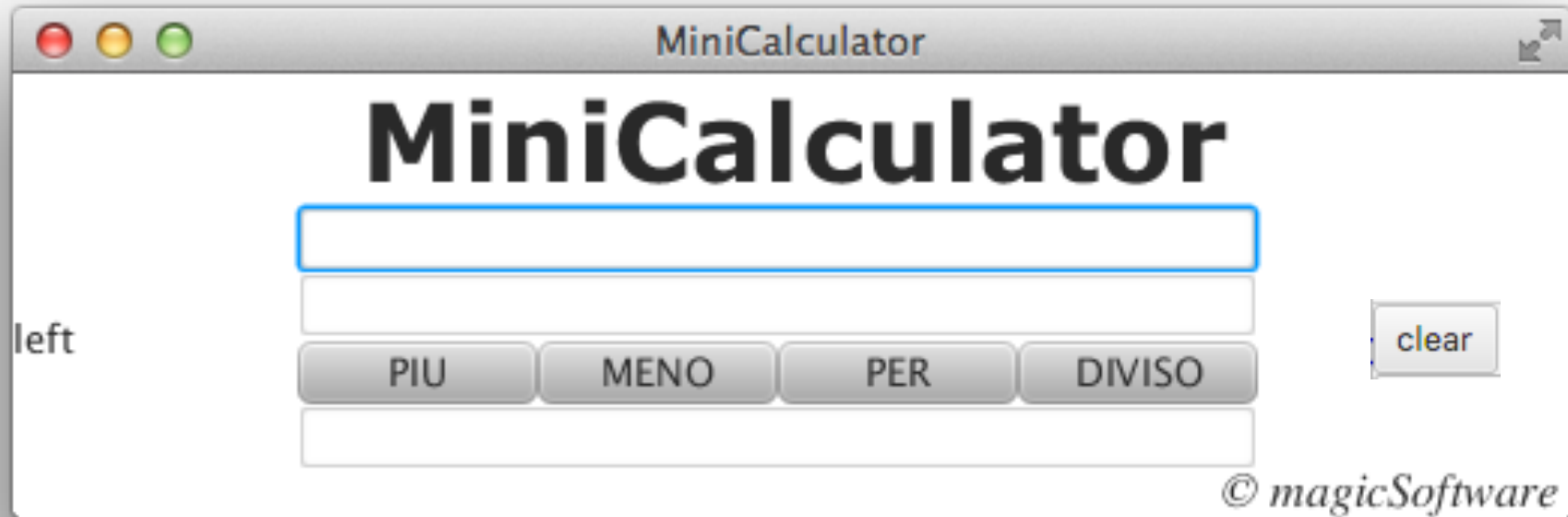
**Incontra la comunità di appassionati
ed esperti del mondo Java™ Virtual
Machine!**

Vola al Java™ User Group Day del Trentino Alto Adige Südtirol ed incontra
persone che si divertono, lavorano con Java™, Kotlin™ e Scala™

Le novità in Java™ 9 -> 12.

Come e perché ho convertito i miei vecchi progetti
Java™ in componenti serverless di prima classe.

Scrivere un minicalcolatore avente l'aspetto presentato dalla seguente immagine (con tutti gli elementi, scritte incluse, posizionate come si vede).



La funzionalità è la seguente:

- I primi due campi (quelli sopra i bottoni centrali) sono editabili dall'utente.
- Il terzo campo (quello sotto i bottoni) non è modificabile dall'utente.
- I 4 bottoni centrali fanno la cosa ovvia: "PIU" somma i due campi superiori, e riporta il risultato nel terzo campo. Analogo comportamento hanno gli altri tre.
- Se alla pressione di uno dei bottoni centrali il contenuto dei due campi superiori non è interpretabile come un floating point, appare un pop-up con un messaggio di errore.
- Il bottone clear ripulisce tutti i campi.

Gestire la pressione di tasti

```
Button b = new Button("PLUS");
EventHandler<KeyEvent> keyEventHandler = new
    EventHandler<KeyEvent>() {
    @Override
    public void handle(KeyEvent e) {
        if (e.getCharacter().equals("+")) {
            System.out.println("Buttom + pressed");
        }
    }
};
b.addHandler(KeyEvent.KEY_PRESSED, keyEventHandler);
```