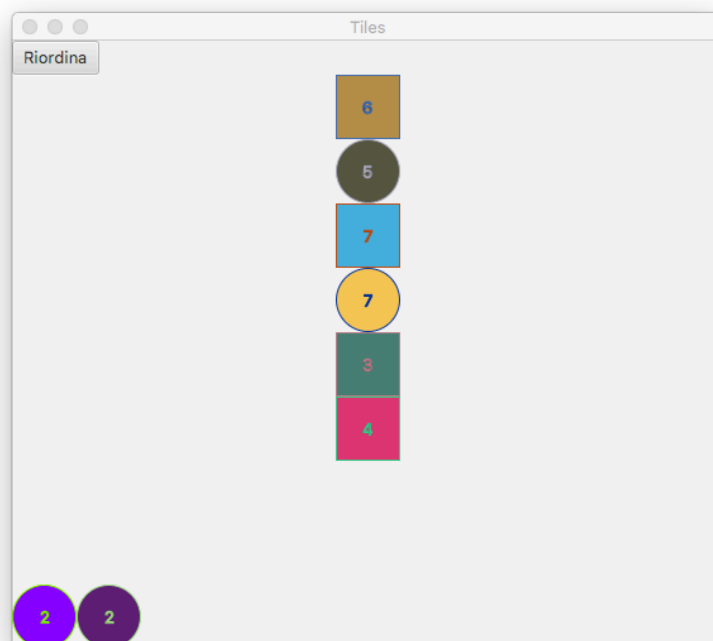


**Compito di Linguaggi di Programmazione (Mod.1)**  
**Prova al calcolatore – giugno 2019**

- 1) Si prepari su un foglio il diagramma UML di tutte le classi coinvolte nel progetto sotto specificato. Non serve mostrare le classi di JavaFX, a parte quelle che hanno un ruolo diretto nel programma, e quelle da cui le classi scritte dal programmatore ereditano direttamente, se ve ne sono. Al termine, il foglio (sul quale devono essere presenti nome, cognome, numero di matricola) dovrà essere consegnato.
- 2) Tile è una componente grafica sulla quale è raffigurato un numero intero compreso tra 0 e 9 (detto “valore”); CircleTile è un Tile di forma circolare; SquareTile è un Tile di forma quadrata.
- 3) Due Tile sono considerati *equivalenti* se sono dello stesso tipo ed hanno lo stesso valore (Attenzione: NON implementare il controllo di equivalenza con la equals!)
- 4) Inizialmente l'utente sceglie tramite una DialogBox (in alternativa, con una penalità in termini di punteggio, da console) un parametro N intero pari ( $3 < N < 11$ ). L'inserimento di valori non validi deve essere intercettato dal sistema, che ripete all'utente la domanda.
- 5) Vengono quindi creati in modo casuale N Tile non equivalenti tra loro: metà dei Tile generati dovranno essere dei CircleTile, e metà degli SquareTile. Il colore di ciascun Tile sarà casuale. (Vi potranno quindi ad esempio essere uno SquareTile ed un CircleTile con valore 4, ma non due SquareTile con valore 4).
- 6) Gli N Tile generati vengono disposti (in ordine di casuale) in una struttura verticale e visualizzati.



- 7) Cliccando sul bottone "Riordina", gli elementi presenti nella struttura verticale vengono riposizionati in ordine numerico (numeri maggiori sotto, numeri minori sopra).
- 8) La pressione da tastiera del tasto R ha lo stesso effetto che premere il bottone Riordina.
- 9) Cliccando su un Tile, il suo numero viene aggiornato: incrementato di 1 per i CircleTile, decrementato di 1 per gli SquareTile, e quindi rimappato nell'intervallo [0-9] (ovvero se il valore raggiunge 10 torna a 1, se raggiunge -1 torna a 9).
- 10) Se dopo l'aggiornamento accade che nella struttura verticale sia già presente un Tile equivalente, entrambi vengono rimossi dalla struttura verticale e "parcheeggiati" in una struttura orizzontale ove rimarranno per sempre. I click su di essi non avranno più alcun effetto.
- 11) Se invece dopo l'incremento non si verifica la condizione di equivalenza suddetta, il Tile viene riposizionato: se è uno SquareTile viene spostato in cima alla struttura verticale, se è un CircleTile va in fondo.
- 12) Premere da tastiera un tasto numerico equivale a cliccare su uno (qualunque) dei Tile che hanno valore pari al numero premuto (opzionalmente su tutti quelli aventi il valore dato).
- 13) Si documenti il codice prodotto con Javadoc.

NOTA: la `getChildren.add()` impedisce di aggiungere oggetti se essi sono equals ad altri già presenti nella lista dei figli. Per questa ragione a punto 3 si sconsiglia di usare la `equals` per controllare se due Tile sono "equivalenti".