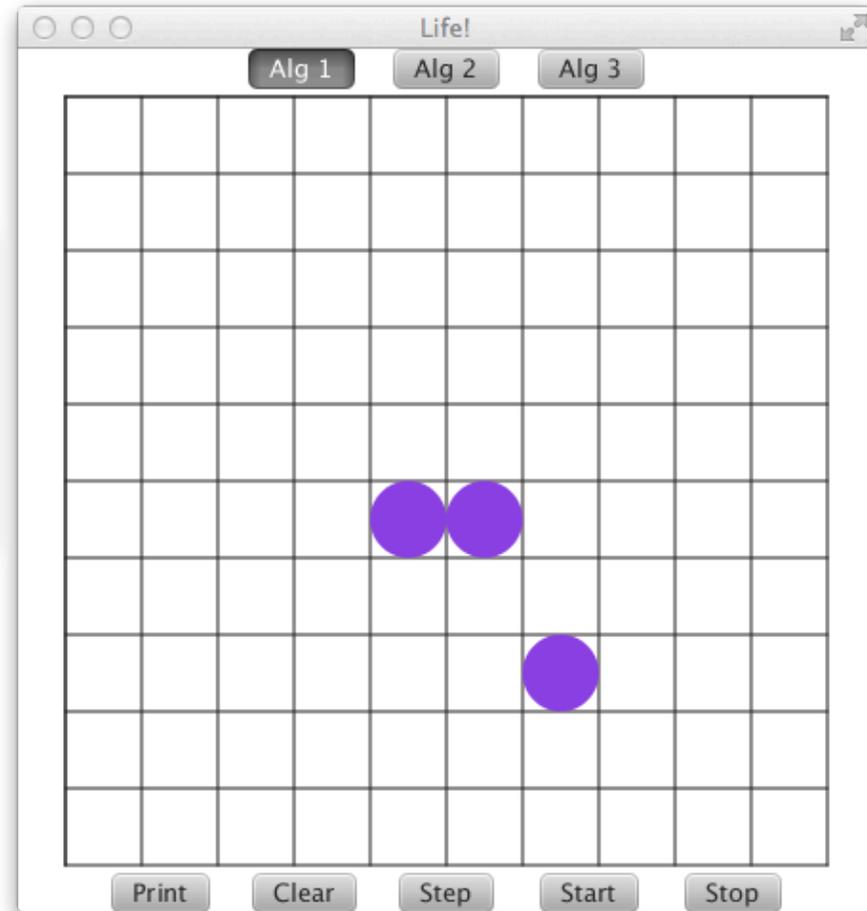




# Statistiche

	Iscritti	Ammessi	Passati	30 +	Amm/Iscr	Pass/Amm	30+/Pass
Giugno	115	50	42	11	43%	84%	26%
Luglio	105	65	53	5	61%	81 %	9,4%
Sett.	47	27	21	2	57%	88%	9,5%
Gennaio	21	10	8	0	47%	80%	0%
Febbraio	35	15	9	0	42%	60%	0%
Totale	182 (323)		127 (167)	18	70 % (pass/iscr. unici) 52% (tentativi con successo)		18%

# Appello di Giugno 2014



# Appello di Giugno '14 – 1-2

1) Scrivere un'applicazione che:

- presenti una griglia NxN (scegliete voi il valore di N che volete – si veda anche il punto 12)
- abbia 8 bottoni, divisi in due gruppi:
  - gruppo1: Print, Clear, Step, Start, Stop
  - gruppo 2: Alg.1, Alg2, Alg3

2) Il codice deve essere documentato con Javadoc.

# Appello di Giugno '14 – 3-4

3) Cliccando su una cella vuota della griglia, apparirà un cerchio. Cliccando su un cerchio, questo scomparirà. (Suggerimento per chi lavora con JavaFX: fare sì che ogni cella della griglia contenga un cerchio, che sarà reso visibile/invisibile giocando sul parametro di Opacità).

4) Premendo il tasto Print, verrà stampata una matrice di occupazione (es. 1 oppure true per le celle occupate, 0 oppure false per le celle libere). La stampa dovrà preferibilmente apparire in una finestra separata, ma è accettabile (con un piccola penalità) che venga fatta in console.

# Appello di Giugno '14 – 5-6

5) Premendo il tasto Clear la griglia verrà svuotata (i cerchi spariranno). Ovviamente cliccando nuovamente su una cella, il corrispondente cerchio riapparirà.

6) Premendo il tasto “Step”, i cerchi si sposteranno. Di default si sposteranno di una cella verso l’alto e a destra (cioè in diagonale).

# Appello di Giugno '14 – 7-8

7) I tre bottoni Alg... sono permettono di scegliere fra tre diversi algoritmi per il movimento dei cerchi: Alg1 corrisponde allo spostamento di default (diagonale verso destra-alto); Alg2 spostamento verso sinistra, Alg3 spostamento verso il basso. L'azione "Step" rifletterà l'algoritmo scelto.

8) E' preferibile che I bottoni Alg1... Alg3 mostrino lo stato, e siano mutuamente esclusivi. Premendone uno, questo resterà "schacciato". Premendone un secondo, il primo tornerà in posizione rialzata ed il secondo apparirà "schacciato".

# Appello di Giugno '14 – 9-12

- 9) Associare i tasti C ed S ai bottoni “Clear” e “Step”
- 10) Il tasto “Start” avvia una animazione che effettuerà uno step ogni secondo.
- 11) Il tasto “Stop” fermerà l’animazione.
- 12) All’avvio il programma mostrerà una prima finestra nella quale si chiederà all’utente di indicare un numero intero. (N) Se l’utente non inserisce un numero, verrà generato un errore. Altrimenti, la finestra si chiuderà, e se ne aprirà un’altra contenente la griglia: l’intero indicato dall’utente determinerà la dimensione della griglia (NxN).

# Package names

- come evitare i conflitti di nomi?

# Node – set/get userData

## setUserData

```
public void setUserData(java.lang.Object value)
```

Convenience method for setting a single Object property that can be retrieved at a later date. This is functionally equivalent to calling the `getProperties().put(Object key, Object value)` method. This can later be retrieved by calling `getUserData()`.

Parameters:

value - The value to be stored - this can later be retrieved by calling `getUserData()`.

## getUserData

```
public java.lang.Object getUserData()
```

Returns a previously set Object property, or null if no such property has been set using the `setUserData(java.lang.Object)` method.

Returns:

The Object that was previously set, or null if no property has been set or if null was set.