

Compagnia aerea

Una compagnia aerea dispone di un **elicottero** che effettua, ogni giorno, voli turistici sulla città. L'elicottero dispone di **N** posti. La compagnia accetta sia **passaggeri singoli** che **gruppi** (di **M** persone con $2 \leq M \leq N$).

I gruppi di passeggeri hanno priorità rispetto ai passeggeri che viaggiano da soli: se ci sono gruppi in attesa di salire sull'elicottero e questi possono entrare, devono avere la precedenza rispetto al passeggero singolo. Tuttavia, se ci sono gruppi in attesa, ma nessuno di questi può entrare a causa del numero di persone (un gruppo non può entrare se salendo il numero massimo **N** di passeggeri caricabili viene sforato), allora un passeggero singolo può entrare.

Una volta che l'elicottero si è riempito, oppure ha aspettato almeno un certo tempo **X** e c'è almeno un passeggero a bordo, il pilota chiude il portello ed il volo ha inizio (durata random). Al termine del volo, dopo l'atterraggio, i passeggeri/gruppi scendono dall'elicottero e si può dunque caricare altri passeggeri.

Si gestisca l'accesso in maniera **FIFO-like**, rispettando cioè, ove possibile, l'ordine di arrivo dei passeggeri/gruppi.

Si implementi la soluzione usando i metodi nativi di Java per la sincronizzazione.