

**Avvertenza.** Non è ammesso l'uso delle classi del package *prog.io* allegato al libro di testo del corso.

## 1 Testo esame

Lo scopo è realizzare il modello di una colonna ascensore, con la simulazione del trasporto di persone.

Le classi da realizzare sono le seguenti (dettagli nelle sezioni successive):

- *Colonna*: muove l'ascensore secondo le richieste da evadere e carica/scarica le persone dall'ascensore
- *Ascensore*: contiene persone, ha una capienza
- *Persona*: ha un peso e un piano a cui vuole andare
- *Piano*: contiene persone che aspettano l'ascensore
- *Main*: contiene il solo metodo *main()* con alcune invocazioni di test (importante è riuscire a testare TUTTI i metodi almeno una volta)

Tali classi dovranno esporre almeno i metodi specificati nelle sezioni seguenti. Eventuali metodi di servizio possono essere aggiunti a piacimento. Ogni classe deve avere il *toString()* che rappresenti lo stato delle istanze e i **costruttori** adeguati per gli attributi che vengono dichiarati. Creare opportunamente i metodi di accesso agli attributi (*set&get*). Si suggerisce, anche dove non segnalato, di utilizzare, se esistenti e se applicabili, le classi parametriche (es. *Vector<E>* invece di *Vector*). Alcuni controlli di coerenza vengono suggeriti nel testo, potrebbero essercene altri a discrezione. I tipi di ritorno possono essere variati (ad es. *boolean* invece di *void* se si vuole ottenere un feedback sul successo dell'operazione) previa autorizzazione del docente. Consiglio di postporre l'implementazione dei controlli di coerenza, fatelo come ultima operazione, prima realizzate un sistema funzionante.

### 1.1 Ascensore

Ha una capienza massima (espressa sia in peso che in nr. di persone trasportabili). Accetta persone in ingresso solo se non ha raggiunto capienza. Deve inoltre esporre i seguenti metodi **pubblici**:

- *boolean sale(Persona p)*: aggiunge una persona all'ascensore, accetta solo se c'è disponibilità di capienza, ritorna *true* se ha successo.
- *void scende(Persona p)*: fa scendere una persona dall'ascensore (attenzione a chi fa scendere, dovete far scendere solo quelli del piano giusto).

### 1.2 Colonna

E' connessa con un Ascensore e con le istanze di Piano. Sposta l'ascensore a seconda delle richieste ai

vari piani (presenza di persone) e fa entrare/uscire le persone nell/dall'ascensore. Tiene traccia del piano a cui e' l'ascensore per poter prendere le persone che attendono a quel piano.

Deve inoltre esporre i seguenti metodi **pubblici**:

- *void muoviAscensore(int prossimoPiano)*: sposta l'ascensore al piano specificato.
- *Piano piano()*: ritorna il piano corrente.
- *int nrPiano()*: ritorna il piano corrente (in numero).
- *void movimentaPersone()*: fa scendere e salire le persone al piano corrente (attenzione che devono scendere le persone che hanno espresso la scelta di questo piano e non altri).

### 1.3 Persona

Ha un peso e un piano di destinazione. Deve inoltre esporre i seguenti metodi **pubblici**:

- *int destinazione()*: ritorna il piano di destinazione.
- *int peso()*: ritorna il peso.

### 1.4 Piano

Contiene le persone che attendono l'ascensore. Deve inoltre esporre i seguenti metodi **pubblici**:

- *int quante()*: ritorna quante persone stanno aspettando.
- *Persona persona(int i)*: ritorna la i-esima persona in attesa.
- *boolean sale(int i, Ascensore a)*: fa salire la i-esima persona in ascensore (se l'ascensore ha capienza) e la rimuove dalla coda (se ha avuto successo), ritorna *true* se ha avuto successo tutto il procedimento.

### 1.5 Main

Deve contenere il metodo *main()* in cui vanno istanziate le classi realizzate e opportunamente testate, ad es. creando almeno un tavolo che ordina alcuni piatti e popolando le istanze per avere una situazione reale a regime, con generazione del conto finale.

## 2 Consegna

Ricordo che le classi devono essere tutte *public* e che vanno consegnati tutti i file *.java* prodotti. NON vanno consegnati i *.class*. Per la consegna, eseguite l'upload dei singoli file sorgente dalla pagina web: <http://upload.dico.unimi.it>.