



Laboratorio di Programmazione

Laurea in Bioinformatica

Web: <http://www.scienze.univr.it/fol/main?ent=oi&id=28023&lang=it>

Docente: *Carlo Drioli*

Email: *drioli@sci.univr.it*

Lucidi a cura di
Nicola Drago Carlo Drioli Federico Fontana

Lezione 18

Sommario

- Esercizi su progettazione di classi e uso di package

Esercizi (1/2)

Si progetti una classe di nome **Treno**, la quale istanzia oggetti in grado di rappresentare la potenza di traino della locomotiva, il numero di vagoni e il peso di un treno. Si assuma che l'oggetto abbia senso solo se il peso del treno non supera la potenza e se il numero di vagoni e' inferiore a 10: quest'ultimo campo dovra' essere pubblicamente accessibile a tutte le classi che intendano adoperare le risorse fornite dalla classe **Treno**

La classe contenga:

- 1) un costruttore **Treno(int p)**, il quale costruisce un oggetto che modella un treno formato dalla sola locomotiva di potenza **p**
- 2) un costruttore **Treno(int p, int n, int m)**, il quale se possibile costruisce un oggetto che modella un treno formato dalla locomotiva di potenza **p** piu' **n** vagoni ciascuno pesante **m**, altrimenti restituisce la sola locomotiva di potenza **p**
- 3) un metodo booleano **accodaVagone(int m)**, che se possibile aggiunge al treno modellato un vagone di massa **m**, e in tal caso restituisce **vero**; altrimenti restituisce **falso**
- 4) un metodo **int sganciaVagone()**, che se possibile elimina il vagone di coda del treno modellato e in tal caso restituisce la sua massa; altrimenti restituisce **0**
- 5) un metodo **int quantoPesa()**, che restituisce il peso del treno modellato
- 6) un metodo **int quantoResta()**, che restituisce il peso che pu• essere ancora aggiunto al treno

Esercizi (2/2)

- 7) un metodo `int quantiVagoni()`, che restituisce il numero dei vagoni che attualmente formano il treno
- 8) un metodo `String toString()`, che restituisce il modello del treno nel formato "`<p>m1:m2:...`", in cui `p` e' la potenza della locomotiva e ciascun elemento tra caratteri due punti contiene la massa `m` del vagone `i`-esimo.

Per modellare i vagoni si consiglia di adoperare un array di interi `mass[i]`, in cui ciascun elemento contiene la massa `m` del vagone `i`-esimo. La classe dovra' far parte di un package di nome `myclasses` accessibile da ambiente Java.

La classe `UsaTreno` contenga un metodo `main` che, appoggiandosi alle risorse della classe `Treno` contenuta nel package `myclasses`, simula la creazione di due treni dotati di locomotive di potenza di traino uguale a 50, a cui sono accodati al piu' 10 vagoni di peso casuale variabile tra 1 e 10.

Il metodo successivamente scambia i vagoni in modo da bilanciare quanto piu' possibile il peso dei due treni secondo una strategia "greedy", ovvero sganciando tutti i vagoni dai due treni e poi assegnando via via il vagone piu' pesante al treno in quel momento piu' leggero.