

# Supereroi

## Progetto di Programmazione del 9 luglio 2018

---

### Introduzione

---

In questo esame vi chiederemo di scrivere un insieme di classi per la gestione di supereroi.

Il progetto verrà valutato prima di tutto in base al suo corretto funzionamento rispetto ai requisiti qui descritti; suggeriamo di commentare con cura il codice, in particolare anche scrivendo i commenti

`javadoc` .

### Realizzazione

---

#### Classe Caratteristica

Le istanze di questa classe rappresentano ciascuna una certa certa caratteristica fisica. Più precisamente, ogni istanza ha un nome (e.g., "altezza", "forza fisica", "numero di orecchie"...), un'unità di misura (e.g., "cm", "N", "#") e un valore normale (il valore che in media quella caratteristica ha per un essere umano ordinario).

La classe ha solo un costruttore

- `public Caratteristica(String nome, String unitaMisura, double valoreNormale)`

e un metodo

- `public double valoreNormale()` : che restituisce il valore normale di questa caratteristica.

La classe inoltre sovrascrive i metodi `toString` , `equals` e `hashCode` di `Object` . Il metodo `equals` deve essere basato solo sul nome. Inoltre esiste un metodo:

- `public double confrontoCon(double v)` che restituisce il rapporto fra il valore `v` e il valore normale della caratteristica.

Ad esempio, la caratteristica

```
orecchie = new Caratteristica("orecchie", "#", 2)
```

rappresenta il numero di orecchie e `orecchie.confrontoCon(4)` restituisce 2.0 perché un essere

con quattro orecchie ha il doppio delle orecchie della norma.

## Classe Supereroe

Le istanze di questa classe rappresentano i supereroi (p.es., Supergirl, Spiderman, Tempest...)

Ogni supereroe ha un nome e una serie di caratteristiche fisiche, a ciascuna delle quali è associato un valore.

La classe ha solo un costruttore

- `public Supereroe(String nome)` : crea un supereroe con il nome indicato.

e un metodo

- `public void addCaratteristica(Caratteristica x, double v)` : aggiunge a questo supereroe la caratteristica `x` con valore `v`.

Si assume che tutte le caratteristiche non esplicitamente aggiunte abbiano il loro valore normale.

Il metodo:

- `public double valore(Caratteristica x)` : restituisce il valore della caratteristica `x` (ricordate che le caratteristiche non esplicitamente aggiunte hanno il valore normale).

Il metodo:

- `public Caratteristica[] superpotereGrande(double t)` : restituisce le caratteristiche di questo supereroe il cui valore sia almeno `t` volte oltre il normale. Ad esempio `superpotereGrande(10)` fornisce le caratteristiche per cui questo supereroe ha un valore almeno dieci volte il normale.

Analogamente, il metodo:

- `public Caratteristica[] superpoterePiccolo(double t)` : restituisce le caratteristiche di questo supereroe il cui valore sia al massimo `t` volte oltre il normale. Ad esempio `superpoterePiccolo(0.1)` fornisce le caratteristiche per cui questo supereroe ha un valore pari a un decimo del normale o meno.

La classe inoltre sovrascrive i metodi `toString`, `equals` e `hashCode` di `Object`.

## Classe Team

Le istanze di questa classe rappresentano team di supereroi.

La classe ha solo un costruttore

- `public Team(String nome)` : crea un team vuoto.

Ci sono poi i metodi:

- `public void add(Supereroe x)` : che aggiunge un supereroe al team
- `public Supereroe[] componenti()` : che restituisce un array con i supereroi che compongono il team.

La classe ha inoltre i seguenti metodi:

- `public Supereroe cercaGrande(Caratteristica x, double t)` : cerca se in questo team c'è un supereroe per cui la caratteristica `x` sia almeno `t` volte il normale; se c'è, ne restituisce uno (qualsiasi); se non c'è, restituisce `null` .

Analogamente c'è un metodo

- `public Supereroe cercaPiccolo(Caratteristica x, double t)` : cerca se in questo team c'è un supereroe per cui la caratteristica `x` sia al massimo `t` volte il normale; se c'è, ne restituisce uno (qualsiasi); se non c'è, restituisce `null` .

Ad esempio:

```
team.cercaGrande(orecchie, 4)
```

cercherà un supereroe che abbia almeno 8 orecchie, mentre (assumendo che l'altezza normale sia 180 cm)

```
team.cercaPiccolo(altezza, 0.01)
```

cerca un supereroe alto al massimo 1.8 centimetri.

## Classe MondoDiSupereroi

Le istanze di questa classe rappresentano mondi di supereroi. Ogni mondo ha un nome, e ad ogni mondo si possono aggiungere squadre di supereroi.

La classe ha solo un costruttore

- `public MondoDiSupereroi(String nome)` : crea un nuovo mondo di supereroi.

e i seguenti metodi:

- `public void add(Team x)` : aggiunge il team `x`
- `public int quantiSupereroi()` : restituisce il numero complessivo di supereroi esistenti in questo mondo: tale numero si può calcolare guardando come è costituito ogni team; **tenete conto del fatto che lo stesso supereroe può appartenere a più di un team, ma non lo dovete contare più**

**volte!**

- `public Team cercaGrande(Caratteristica x, double v)` : restituisce un qualsiasi team di questo mondo che contenga almeno un supereroe la cui caratteristica `x` sia almeno `v` volte la norma
- `public Team cercaPiccolo(Caratteristica x, double v)` : restituisce un qualsiasi team di questo mondo che contenga almeno un supereroe la cui caratteristica `x` sia al massimo `v` volte la norma.