

Si lascia allo studente la definizione dell'azione (coerente) da svolgere nel caso in cui un controllo non vada a buon fine; una stampa può essere sufficiente.

### Esercizio 1

Si scriva il codice di una classe Java **P** che implementi una **particella elementare**, e che mantenga l'informazione sulla **velocità media** in km/s, che deve essere un valore maggiore di zero.

Si prevedano due **costruttori** per **P**: uno che inicializzi le istanze con il valore della velocità media passato come parametro, controllando che sia maggiore di zero; l'altro che inicializzi le istanze alla velocità di default di 300.000 km/s.

Nella classe **P** si devono prevedere due operazioni: una di **distanza** che accetta come parametro una quantità di tempo in secondi e restituisce la distanza percorsa in km; una di **modifica** che permetta di impostare una nuova velocità media, sempre facendo un controllo che la nuova velocità sia conforme alle specifiche.

### Esercizio 2

Si scriva il codice di una classe Java **PQ** che estenda **P**, e che implementi una **particella quantistica**, la cui posizione è data con una certa probabilità; PQ deve mantenere una informazione che rappresenta, in percentuale, la **probabilità** che la particella si sia spostata; tale valore deve essere non negativo e ovviamente inferiore al 100%.

Si preveda un **costruttore** che inicializzi le istanze di **PQ** con il valore della probabilità passato come parametro, sempre controllando che sia un valore corretto, e un costruttore che inicializzi le istanze di **PQ** con il valore di default corrispondente al 50%.

Rispetto alla superclasse, si deve **sovrascrivere** l'operazione di **distanza** per calcolare una variabile random **r** (tramite la funzione `Math.random()` che restituisce un double comprese tra 0.0 e 1.0 escluso) e restituire la distanza percorsa se **r** è *maggiore o uguale* alla probabilità impostata, oppure restituire 0 se **r** è *minore* della probabilità impostata.

Inoltre, la classe **PQ** aggiunge una operazione di **modifica** della probabilità, sempre controllando che sia un valore corretto.

### Esercizio 3

Scrivere un metodo main che usi le classi precedenti e che metta in evidenza almeno un **comportamento polimorfico**. Si **spieghi** qual è l'effetto del polimorfismo nel codice scritto.