

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 **C** che implementi un componente per calcolare il **conto** di una serie di acquisti, mantenendo l'informazione sul **conto parziale**.

Si preveda un **costruttore** per **C** che inicializzi le istanze con il valore del conto parziale a zero.

Nella classe **C** si devono prevedere tre operazioni: una di **lettura** che restituisce il valore del conto parziale; una di **aggiunta** che accetta come parametro il costo di un acquisto e, dopo aver verificato che sia un valore non negativo, la aggiunge al conto parziale; una di **annullamento** che permetta di azzerare il conto parziale.

Esercizio 2

Si scriva il codice di una classe Java **Ct** che estenda **C**, e che implementi un componente per calcolare il costo aggiungendo le **tasse locali**; **Ct** deve mantenere una informazione che rappresenta, in percentuale, le **tasse** da aggiungere; tale valore è lo stesso per tutti gli acquisti e deve essere non negativo.

Si preveda un **costruttore** che inicializzi le istanze di **Ct** con il valore delle tasse passato come parametro, controllando che sia un valore corretto, e un costruttore che inicializzi le istanze di **Ct** con il valore di default delle tasse del 21%; entrambi i costruttori devono inicializzare il conto parziale a zero.

Rispetto alla superclasse, si deve **sovrascrivere** l'operazione di **lettura** per restituire il costo parziale incrementato delle tasse locali.

Inoltre, la classe **Ct** aggiunge una operazione di **modifica** delle tasse, sempre controllando che il nuovo valore sia conforme alle specifiche.

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.