

Esercizi di programmazione

Ereditarietà

1. Migliorare il metodo `addInterest` della classe `SavingAccount` per calcolare l'interesse in base al saldo minimo che è stato raggiunto nel periodo intercorso dall'ultima chiamata di `addInterest` (suggerimento: modificare il metodo `withdraw` e aggiungere una variabile istanza per registrare il saldo minimo).
2. Aggiungere la classe `TimeDepositAccount` nella gerarchia dei conti bancari. Il conto di deposito a tempo è identico a un conto di risparmio, a parte il fatto che si promette di lasciare il denaro nel conto per un fissato numero di mesi ed è prevista una penale per i prelievi anticipati. Costruire un conto con un tasso di interesse e un numero di mesi per la scadenza. Nel metodo `addInterest` decrementare il conteggio dei mesi. Se il conteggio è positivo nel corso di un prelievo, applicare la penale.
3. Implementare una sottoclasse `Square` che estenda la classe `Rectangle`. Il costruttore riceve le coordinate `x` e `y` del centro e la lunghezza del lato del quadrato. Chiamare i metodi `setLocation` e `setSize` della classe `Rectangle`. Fornire un metodo `getArea` per calcolare e restituire l'area del quadrato. Scrivere una classe di test.
4. Implementare una superclasse `Person`. Creare due classi, `Student` e `Instructor`, che ereditano da `Person`. Una persona possiede un nome e un anno di nascita, uno studente ha una disciplina di specializzazione e un istruttore un corso da svolgere. Scrivere le definizioni delle classi, il costruttore e il metodo `toString` per tutte le classi. Fornire una classe di Test.
5. Scrivere una superclasse `Worker` e le sottoclassi `HourlyWorker` e `SalariesWorker`. Ciascun lavoratore ha un nome e un regime retributivo. Scrivere un metodo `computePay(in hours)` per calcolare la paga settimanale di ciascun lavoratore. Un lavoratore pagato a ore riceve la paga oraria calcolata sul numero effettivo di ore lavorate fino ad un massimo di 40. Se si superano le 40 ore, le ore eccedenti sono retribuite una volta e mezzo. Il lavoratore pagato settimanalmente riceve la paga oraria corrispondente a 40 ore, indipendentemente dal numero di ore lavorate effettive. Fornire un programma di Test che usa il polimorfismo per collaudare le classi e i metodi.

Interfacce e polimorfismo

1. Definire una classe `Quiz` che realizzi l'interfaccia `Measurable`. Un quiz ha un punteggio e un voto in lettere (ad esempio B+). Usare l'implementazione della classe `DataSet` per elaborare una raccolta di quiz. Visualizzare il punteggio medio e il quiz con punteggio massimo, sia in lettere che numeri.
2. Definire una classe `Pearson`. Una persona ha un nome e un'altezza in centimetri. Usare l'implementazione della classe `DataSet` per elaborare una raccolta di oggetti di tipo `Pearson`. Visualizzare l'altezza media e il nome della persona più alta.
3. Modificare l'implementazione della classe `DataSet` in modo che calcola anche l'elemento di misura minima.