
Esercizi: ereditarietà e riutilizzo del codice

Esercizio

- Implementare (riutilizzando il codice in comune) e collaudare le classi:
 - Personale
 - ha un nome e una retribuzione
 - Impiegato
 - ha un nome, una retribuzione, un incarico
 - Manager
 - ha un nome, una retribuzione, un incarico, un'area di competenza
- Per ogni classe implementare i metodi toString, equals e clone

Esercizio

- Considerare un'azienda con lavoratori retribuiti a ore e lavoratori retribuiti con un fisso settimanale.
- Per i lavoratori a ore, la paga settimanale è calcolata moltiplicando la paga oraria per il numero di ore effettivamente lavorate fino ad un massimo di 40 ore, se le ore lavorate eccedono le 40 ore, allora la parte di ore eccedente 40 viene retribuita una volta e mezzo la paga oraria ordinaria.
- I lavoratori retribuiti con un fisso ricevono la paga di 40 ore indipendentemente dalle ore effettivamente lavorate.
- Progettare le classi necessarie per il calcolo delle paghe settimanali
- Scrivere un programma di test che crei un elenco di 20 lavoratori (determinando casualmente il tipo di retribuzione), permetta di inserire per ogni lavoratore il numero di ore lavorate e dia in output per ogni lavoratore il nome e la sua paga settimanale
- Riutilizzare il codice dove possibile

Esercizio

- Implementare i metodi toString, equals e clone per le tre classi
 - BankAccount
 - SavingsAccount
 - CheckingAccount

Esercizio

- Fornire un programma di prova per il collaudo delle tre classi create
 - Creare un oggetto s1 di tipo SavingsAccount con interestRate = 0.5 e un oggetto di tipo CheckingAccount c1 con balance = 0
 - Clonare i due oggetti, ottenendo s2 e c2
 - Stampare il contenuto di tutti e quattro gli oggetti
 - Invocare il metodo equals per verificare che s1 e s2 sono uguali e che c1 e c2 sono uguali
 - Depositare e prelevare 100 euro da c2
 - Invocare il metodo equals per verificare che c1 e c2 sono diversi
 - Saldi uguali, ma numero di transazioni diverso
-

Esercizio

- Definire una superclasse astratta `FiguraGeometricaPiana` e le sottoclassi
 - `Cerchio`, `Rettangolo` e `TriangoloIsoscele`
- Per ciascuna classe, definire
 - le variabili di istanza per le dimensioni che caratterizzano il tipo di figura,
 - il costruttore per la creazione di un'istanza di ciascuna figura date le dimensioni,
 - due metodi `getArea()` e `getPerimetro()`, che restituiscono rispettivamente l'area e il perimetro della figura

Esercizio

- Scrivere un programma di Test per le classi create nell'esercizio precedente:
 - Creare un array di figure geometriche, contenente cinque oggetti per ognuna delle classi create;
 - Stampare la figura con l'area massima, quella con l'area minima, quella il perimetro massimo e quella con il perimetro minimo tra tutte le figure presenti nell'array

Esercizio

- Progettare un semplice sistema bancomat utilizzando il procedimento studiato per la progettazione orientata agli oggetti. Il sistema bancomat deve consentire l'esecuzione delle operazioni principali su di un conto bancario.
- I conti bancari possono essere di tre tipi:
 - conto corrente,
 - libretto (conto) di deposito e
 - carta di credito.