

Sistemi di Elaborazione 2

Esercizi – Seconda lezione di laboratorio

Esercizio n. 1

Sviluppare e realizzare una classe `Account` che rappresenta un conto corrente bancario. L'oggetto `Account` deve avere tre attributi: nome del conto, nome intestatario e saldo corrente. La classe deve prevedere i seguenti comportamenti per costruttori e metodi:

- `Name()`: configura un nuovo conto con nome "name", intestatario "owner" e saldo 0.
- `Name(String s, String t)`: configura un nuovo conto con nome `s`, intestatario `t` e saldo 0.
- `Name(String s, String t, double n)`: configura un nuovo conto con nome `s`, intestatario `t` e saldo `n`.
- `void deposit(double n)`: aggiunge l'importo `n` al saldo corrente del conto associato.
- `void withdraw(double n)`: sottrae l'importo `n` dal saldo corrente del conto associato.
- `void close()`: azzerà l'importo e imposta l'intestatario su "".
- `String getAccountName()`: restituisce il nome del conto associato.
- `String getOwner()`: restituisce l'intestatario del conto associato.
- `double getBalance()`: restituisce il saldo del conto associato.
- `void setName(String s)`: imposta il nome del conto associato su `s`.
- `void setOwner(String s)`: imposta l'intestatario del conto associato su `s`.
- `String toString()`: restituisce una rappresentazione testuale degli attributi del conto associato. [Cohon, Davidson, es. 4.24, modificato]

Scrivere un `main` che verifica le funzionalità della classe.

Nota 1: in genere è opportuno dichiarare gli attributi delle classi con il modificatore `private`, in modo che non sia possibile l'accesso diretto agli attributi stessi. E' necessario quindi prevedere dei metodi che permettano di impostare (`set`) e ottenere (`get`) il valore di tali attributi. Lo scopo dell'incapsulamento è garantire la coerenza e la correttezza dei dati. Nell'esempio considerato, è opportuno controllare che il nome del conto e dell'intestatario siano diversi dalla stringa vuota ("") e da `null`, e che il valore `n`, argomento dei metodi `deposit` e `withdraw` sia maggiore di zero.

Nota 2: si noti che, come visto in aula, il confronto di uguaglianza con la stringa vuota va svolto mediante il metodo `equals()`, mentre il confronto con `null` va svolto mediante l'operatore `'=='`.

Nota 3: nel caso il controllo di correttezza sui dati passati al costruttore e ai setter fallisca, il metodo deve lanciare una eccezione di tipo `InvalidValueException`, che va opportunamente dichiarata.

Esercizio n. 2

Impostare una classe `Nome` che rappresenta dei nomi. Ogni oggetto `Nome` deve avere due attributi: nome e cognome. La classe deve avere un costruttore predefinito e un costruttore specifico; deve inoltre offrire supporto per l'incapsulamento dei dati attraverso `getter` e `setter` (`getNome()`, `setNome()`, ...). La classe deve prevedere un metodo `toString()` per produrre il nome completo. Effettuare un opportuno controllo sulla correttezza dei dati passati al costruttore e ai setter.

Scrivere un `main` che verifica le funzionalità della classe.