

## Prova C – 18 Luglio 2013 – Gestore Banca

Si vuole progettare un sistema per la gestione dei conti correnti di una banca.

Definire le classi per la gestione di semplici operazioni bancarie: prelievo, versamento e bonifico. Queste operazioni hanno dei dati comuni: la data dell'operazione (una `Date` dal package `prog.util`), il numero di conto corrente (un intero positivo) e l'importo dell'operazione (un `float` positivo). La classe `Prelievo` deve gestire il dato che indica da dove è stato effettuato il prelievo (una stringa, ad esempio "bancomat"). La classe `Versamento` deve gestire il dato che indica il mezzo usato per il versamento che può essere `contanti` o assegno o altro mezzo. Ha anche un dato opzionale che riporta la causale del versamento (una stringa) che essendo opzionale non deve essere inizializzata nel costruttore ma dovrà esserci un metodo `setCausale` per fare ciò. La classe `Bonifico` deve gestire il dato che indica il beneficiario del bonifico (una stringa, ad esempio le coordinate bancarie del beneficiario), la data in cui verrà eseguito e un dato opzionale per la causale del bonifico che deve essere trattato come la causale della classe `Versamento`. Per tutte le operazioni, definire un metodo `getCC` che ritorna il numero di conto corrente, un metodo `getImporto` che ritorna l'importo e un metodo `stampa` che stampa la descrizione dell'operazione. Metti tutti gli opportuni controlli e solleva le eccezioni se necessario.

Ogni operazione ha un costo: il prelievo è fisso e pari a 1 euro, il versamento per contanti è gratis, mentre quella con assegno è 1 euro, altro mezzo è 5 euro. Il bonifico costa 1 euro se la data è entro 2 giorni, se è maggiore è gratis.

Definire una classe `Conto` per gestire i dati dei conti correnti: il numero del conto (un intero positivo univoco), l'intestatario (una stringa), il saldo (un `float` positivo, negativo o zero) e la lista di tutte le operazioni effettuate sul conto corrente. Il costruttore prende come parametri l'intestatario, mentre il numero di conto viene creato in modo univoco. Il saldo è inizializzato a zero.

Definisci una classe `banca` che ha le seguenti operazioni

- apertura conto corrente – crea un conto corrente e lo aggiunge alla lista dei conti e lo restituisce
- esegui operazione che prende in input un oggetto `Operazione` e se rappresenta una operazione relativa ad un conto corrente valido aggiunge l'operazione alla lista e aggiorna il saldo tenendo conto anche del costo dell'operazione.
- stampa movimenti che prende in input un numero di conto corrente e un intero `n` e stampa le descrizioni delle ultime `n` operazioni (in ordine di data dalla più recente) e il saldo per quel conto corrente.
- Lettura da file: legge i conti correnti (per semplicità non le operazioni, solo intestatario, numero e saldo) da file e li carica in memoria.
- Stampa saldi: stampa il saldo di tutti i conti correnti in ordine di saldo decrescente.

All'incirca ogni funzionalità richiede la scrittura di un metodo a cui passare i parametri opportuni. Metti le funzionalità maggiori in una classe "Banca". Non è necessario fare alcun input con l'utente. Scrivi però un main in cui testi **tutti** i metodi sopra descritti più eventuali altri per assicurarti che funzionino. Nel main non chiedere nulla all'utente ma mostra solo i risultati dei passi. Crea anche un file di conti d'esempio e prova caricarlo.

**Istruzioni pratiche:** usa il desktop come workspace di eclipse e crea un progetto con il tuo cognome.nome: in questo modo ti troverai già sul desktop una cartella che contiene tutto quello che serve per la valutazione. Crea un file `readme.txt` (in cui spieghi la tua applicazione) con eclipse nel default package. Anche il file dei conti mettilo nella root del progetto, per riferirti ad esso basterà che usi il suo nome (senza path).