Metodologie di Programmazione
Prova d'Esame (18/09/2012) – A.A. 2011/2012
Proff. Roberto Navigli e Francesco Parisi Presicce

Nome _____ Cognome _____

Le soluzioni devono essere scritte in modo **chiaro** e **ordinato**. Il codice deve essere **indentato** per facilitarne la lettura. Ogni esercizio ha un punteggio massimo. Il voto (in trentesimi) sarà determinato dalla somma dei punteggi ottenuti nei vari esercizi.

Esercizio 1 (max 15 punti). Si vuole realizzare un insieme di classi per la gestione dei conti di più banche italiane.

[1 pt] Definire una classe `Banca` contenente i dati della banca (come campi necessari e immutabili): il nome esteso della banca, la sigla/acronimo della banca, una stringa costante contenente i due caratteri del codice nazionale italiano IT, il codice ABI associato alla banca (5 cifre). Inoltre, ogni banca ha associata una lista di filiali, istanze della classe `Filiale` (accessibile solo attraverso i metodi che specificheremo dopo).

[1 pt] Definire la classe `Filiale`. Ogni istanza di `Filiale` deve contenere (come campi necessari e immutabili): l'indirizzo della filiale (via e civico), la città e il CAP, il codice CAB (5 cifre), e un riferimento all'istanza della banca della filiale. Inoltre, ogni filiale ha associata una lista di conti, istanze della classe `Conto` (accessibile solo attraverso i metodi che specificheremo dopo).

[1 pt] Definire la classe `Conto` contenente (come campi necessari e immutabili): un codice cliente di 12 cifre che individua univocamente l'intestatario del conto, un riferimento all'istanza di `Filiale` dove il conto è aperto, il numero del conto (12 cifre alfanumeriche), un codice di controllo di un carattere necessario per la costruzione del codice BBAN del conto (vedi dopo), un codice di controllo di due cifre necessario per il calcolo dell'IBAN (vedi dopo). Inoltre, devono essere presenti il saldo iniziale del conto e una lista di operazioni, istanze della classe `Operazione`.

Si scrivano i seguenti metodi della classe `Conto`:

[1 pt]
- un metodo che costruisce e restituisce il codice BBAN (il codice BBAN è una stringa di 23 lettere formata dalla lettera del codice di controllo, ABI, CAB, e numero del conto corrente);

[1 pt]
- un metodo che costruisce il codice IBAN (formato dal codice nazionale, dalle due cifre del codice di controllo e dal BBAN);

[3 pt] La classe `Operazione` è una classe non istanziabile che contiene un campo (necessario e immutabile) pari al valore in euro dell'operazione (un valore sempre positivo) e la data dell'operazione (usare la classe `java.util.Date`). Definire i metodi per l'accesso a questi campi. Questa classe deve avere il costruttore necessario a inizializzare i suoi campi, e deve contenere un metodo astratto che restituisce l'importo positivo/negativo dell'operazione (positivo se i soldi vanno aggiunti al conto, negativo se i soldi vanno sottratti).

[2 pt] La classe `Operazione` ha due sottoclassi: `Prelievo` e `Versamento` per le corrispondenti operazioni. Definire il costruttore di queste classi concrete (riusando quello della superclasse). Implementare la parte astratta di `Operazione`.

[2 pt] Definire un metodo della classe `Banca` che riceve in input un codice cliente e restituisce la lista dei conti associati al cliente.

[3 pt] Definire un metodo della classe `Conto` che riceve come input una data iniziale e una data finale e restituisce un estratto del conto (modellato mediante l'apposita classe `Estratto`) contenente la lista ordinata per data crescente delle operazioni del conto effettuate nel periodo specificato. Se la data di inizio dell'estratto è maggiore di quella di fine, allora il costruttore deve emettere una eccezione non controllata di tipo `PeriodoEstrattoNonValidoException`.

Esercizio 2 (max 5 punti). Definire un metodo che prende in ingresso il nome di un file sorgente, il nome di un file destinazione e due caratteri e copia il file sorgente di testo nel file destinazione rimpiazzando tutte le occorrenze del primo carattere con il secondo carattere. Prevedere le opportune eccezioni.

Esercizio 3 (max 8 punti). Definire un metodo generico che prende in input una matrice e crea e restituisce una lista di liste che rappresenta la matrice. Ogni sottolista contiene gli elementi della corrispondente riga della matrice di input. La matrice può avere righe di lunghezza differente. Il tipo restituito non deve dipendere dalla particolare implementazione usata per le liste.

Esercizio 4 (max 5 punti). Il seguente codice Java contiene uno o più errori. Trovare gli errori e spiegarli. In particolare, dire per ogni errore se si verifica in compilazione o durante l'esecuzione.

```
import java.util.*;

public class Test
{
    public static <T> void copy(T[] a, List<T> list)
    {
        int n = (a.length < list.size() ? a.length : list.size());
        for (int i = 0 ; i < n ; i++)
            a[i] = list.get(i);
    }
    public static void main(String[] args)
    {
        List<Number> nL = new ArrayList<Number>();
        nL.add(0.33);
        Integer[] intA = new Integer[10];
        copy(intA, nL);
        Object[] objA = new Object[5];
        copy(objA, nL);
        nL.clear();
        nL.add(13);
        copy(intA, nL);
    }
}
```

Esercizio 5 (max 5 punti). Scrivere un metodo generico e ricorsivo che, data in ingresso una lista di valori di tipo comparabile (ad esempio, stringhe o interi), restituisca true se ciascun elemento in posizione dispari è minore dell'elemento successivo in posizione pari e se ciascun elemento in posizione pari è maggiore dell'elemento successivo in posizione dispari. Ad esempio, il metodo restituisce true per la sequenza 4, 7, 1, 2, 1, 5, 3 e false per la sequenza 4, 7, 8, 1.