

Cognome \_\_\_\_\_ Nome \_\_\_\_\_  
Matricola \_\_\_\_\_ Postazione PC \_\_\_\_\_

Corso di Laurea in Ingegneria Gestionale

Esame di Informatica - a.a. 2012-13

15 gennaio 2014

### Testo

Il database di una banca è costituito da due vettori paralleli. Il primo è denominato “clienti” e contiene oggetti di tipo “Correntista” che rappresentano i correntisti presenti nell’archivio della banca. Il secondo vettore è denominato “conti” e contiene oggetti di tipo “Conto\_corrente” che rappresentano le informazioni di un conto corrente di un cliente. Ogni cliente può avere più di un conto corrente nella banca, in quel caso le informazioni del cliente saranno replicate.

Per ogni correntista presente nella posizione *i*-esima del vettore “clienti” si troveranno le informazioni relative al suo conto corrente nella corrispondente posizione del vettore “conti”. Nel caso che il correntista in posizione *i*-esima non abbia alcun conto aperto, nella sua posizione nel vettore “conti” sarà presente un riferimento *null*. Entrambi i vettori hanno dimensione pari alla costante “MAX\_CLIENTI” (inizializzata a 1024). Se il numero di correntisti contenuti nell’archivio è inferiore a “MAX\_CLIENTI”, i primi elementi del vettore conterranno gli oggetti di tipo “Correntista”, mentre gli altri conterranno riferimenti *null*. Tutti gli elementi *null* del vettore “clienti” si devono trovare alla fine del vettore e non possono trovarsi in mezzo agli elementi validi.

Le classe Correntista contiene le informazioni relative ad un correntista ed un metodo per stampare queste informazioni:

```
public class Correntista {
    public int id;
    public String nome;
    public String cognome;
    public int annoNascita;
    public int meseNascita;
    public int giornoNascita;

    public Correntista(int id, String nome, String cognome, int giornoNascita,
        int meseNascita, int annoNascita) {
        this.id = id;
        this.nome = nome;
        this.cognome = cognome;
        this.giornoNascita = giornoNascita;
        this.meseNascita = meseNascita;
        this.annoNascita = annoNascita;
    }

    public String toString(){
        return "[" + this.id + "] " + this.nome + " " + this.cognome + " \t"
            + this.giornoNascita + "/" + this.meseNascita + "/" + this.annoNascita;
    }
}
```

La classe Conto\_corrente contiene le informazioni relative al conto corrente di un correntista.

```
public class Conto_corrente {
    static private int numeroProgressivo = 0;
    private int numeroConto;
    public double saldo;
    public int annoApertura;
    boolean mutuo;
    public String filiale;

    public Conto_corrente(double saldo, int annoApertura, String filiale, boolean mutuo) {
        this.numeroConto = numeroProgressivo++;
        this.saldo = saldo;
        this.annoApertura = annoApertura;
        this.filiale = filiale;
        this.mutuo = mutuo;
    }

    public String toString() {
        return Integer.toString(numeroConto);
    }
}
```

**Si consiglia di procedere implementando un metodo e successivamente la parte del main che utilizza tale metodo.** Le varie operazioni devono essere eseguite sulla porzione significativa dell'archivio, cioè la porzione di "clienti" che non contiene riferimenti "null".

**A) Scrivere il metodo statico:**

```
public static int numEntrySenzaConto(Correntista[] clients, Conto_corrente[] cc_db)
```

Il metodo deve effettuare una ricerca tra il 3° ed il 10° elemento del vettore "clients" e restituire il numero di elementi presenti nel suddetto vettore a cui non è associato nessun conto nel vettore "cc\_db".

**B) Scrivere il metodo statico:**

```
public static void ordinaClienti(Correntista[] clients, Conto_corrente[] cc_db)
```

Il metodo deve ordinare, nel vettore "clients", gli elementi in maniera crescente usando come criterio la giustapposizione di giorno, mese ed anno di nascita, interpretata come int. Ad esempio se la data di nascita è 15/10/1914, il numero da considerare per l'ordinamento è 15101914. Suggerimento: una stringa può essere convertita in intero tramite la funzione di libreria `int Integer.parseInt(String tmp)`.

**C) Scrivere il metodo statico:**

```
public static void arrotondaSaldo(Correntista[] clients, Conto_corrente[] cc_db, String cognome)
```

Il metodo deve arrotondare il saldo per ogni conto associato ai correntisti il cui cognome corrisponde a quello passato come parametro. L'arrotondamento deve essere effettuato sempre all'intero più vicino (quello inferiore se la parte decimale è minore di 0.5, quello superiore se è maggiore o uguale). Si suggerisce di utilizzare per lo svolgimento la funzione di libreria `double Math.floor(double d)` che restituisce l'intero inferiore di `d` espresso come `double`.

**D) Scrivere il metodo statico:**

```
public static boolean delCorrentistiAnno(Correntista[] clients, Conto_corrente[] cc_db, int mese1, int mese2)
```

Il metodo deve eliminare dal database specificato dai parametri "clients" e "cc\_db" tutti i conti di correntisti nati tra il mese1 ed il mese2, estremi inclusi. Se mese1 e mese2 non sono parametri validi, cioè non compresi tra 1 e 12, il metodo non deve effettuare eliminazioni. Se sono entrambi validi ma mese1 è maggiore di mese2 li si dovrà ordinare. Il metodo restituisce *true* se è stata effettuata almeno una eliminazione, *false* altrimenti. Anche i rispettivi correntisti devono essere eliminati (lasciando l'archivio in uno stato consistente).

**E) Scrivere il metodo main che:**

- definisca ed inizializzi i vettori "clienti" e "conti" secondo i valori riportati in tabella e stampi a video l'archivio. La stampa consiste nello stampare le informazioni di ogni cliente (usando il metodo "toString" della classe "Correntista") e, se il correntista possiede un conto, deve essere indicato anche (e solo) il numero di conto e il saldo.

Id	Nome e Cognome	Data di Nascita	Saldo	Anno Apertura	Filiale	Mutuo
0	Mario Rossi	15/11/1914	0	1999	Roma1	No
1	Piero Rossi	1/10/1958				
0	Mario Rossi	15/11/1914	2500	2009	Roma1	Sì
2	Giovanni Verdi	22/8/1980	500,15	2007	Milano1	Sì

- Stampi un messaggio che indichi il numero di correntisti senza alcun conto, utilizzando il metodo del punto A.
- Ordini l'intero archivio utilizzando il metodo del punto B e stampi a video l'archivio prima e dopo l'ordinamento.
- Utilizzando il metodo C, aggiorni l'archivio per i correntisti il cui cognome corrisponde a "Verdi". Al termine viene stampato l'archivio aggiornato.
- Elimini i conti corrente dei correntisti nati tra i mesi [3,14] e tra i mesi [Luglio, Ottobre], utilizzando il metodo del punto D. Al termine di ogni operazione avvenuta con successo viene stampato l'archivio aggiornato.