

# ESAME DI FONDAMENTI DI INFORMATICA T-2 del 13/06/2012

Proff. E. Denti – G. Zannoni

Tempo a disposizione: 4 ore MAX

**NB: il candidato troverà nell'archivio ZIP scaricato da Esamix anche il software "Start Kit"**

## **NOME PROGETTO ECLIPSE: CognomeNome-matricola (es. RossiMario-0000123456)**

La società elettrica *ElettroDent* ha richiesto lo sviluppo di un'applicazione che permetta ai potenziali utenti di simulare il costo della bolletta elettrica con le proprie mirabolanti tariffe.

### DESCRIZIONE DEL DOMINIO DEL PROBLEMA.

La società offre sia tariffe *a consumo*, sia tariffe *flat* che comprendono già una certa quota di KWh mensili.

Nelle **tariffe a consumo**, l'importo da pagare dipende dai KWh effettivamente consumati nel mese. Il prezzo del singolo KWh può essere o sempre uguale, o differenziato in base ai KWh consumati (ovvero il prezzo cambia quando si supera una certa soglia di KWh consumati nel mese). Inoltre, il prezzo del KWh va sempre incrementato di una quota che rappresenta i costi di rete, di distribuzione, etc, che supporremo pari al 50% del costo dell'energia. Al totale così calcolato dovranno poi essere aggiunte le tasse (accise e IVA), come più oltre precisato. Ogni tariffa specifica:

- il nome dell'offerta commerciale
- il prezzo base (Eurocent/KWh)
- un eventuale prezzo diverso (sempre in Eurocent/KWh) da applicare oltre una certa soglia di KWh/mese.

In formule:

$$\text{costoMensile} = 1,5 * \begin{cases} \text{KWh} * \text{prezzoBase}, & \text{se prezzo dei KWh è unico o il consumo è sotto soglia} \\ \text{soglia} * \text{prezzoBase} + (\text{KWh} - \text{soglia}) * \text{prezzoExtra} & \text{se vi sono due prezzi differenziati} \end{cases}$$

Nelle **tariffe flat**, l'importo da pagare ogni mese è fisso, purché i KWh consumati restino entro una soglia prestabilita; eventuali KWh non usati nel mese non vengono rimborsati e non possono essere usati il mese successivo. Al contrario, se si consumano più KWh di quelli compresi nell'offerta, i KWh eccedenti la soglia devono essere pagati come extra, al prezzo specificato. Anche qui, al totale dovranno essere aggiunte le tasse (accise e IVA), come più oltre precisato.

Ogni tariffa flat specifica:

- il nome dell'offerta commerciale
- la soglia mensile prestabilita (KWh compresi nell'offerta) e il relativo importo fisso (Euro);
- il prezzo per gli eventuali KWh consumati oltre la soglia (Eurocent/KWh).

In formule:

$$\text{costoMensile} = \begin{cases} \text{importoFisso}, & \text{se i KWh consumati nel mese sono entro la soglia} \\ \text{importoFisso} + (\text{KWh} - \text{soglia}) * \text{prezzoExtra} & \text{se vi sono consumi oltre la soglia} \end{cases}$$

**Le tasse**, da aggiungere sempre al costo mensile, comprendono due voci:

- le **accise**, calcolate in proporzione ai KWh consumati (2,27 Eurocent/KWh), esclusi i primi 150 KWh mensili;
- l'**IVA** (10%), applicata sul subtotale precedente, accise incluse.

Il file di testo [Tariffe.txt](#) contiene la descrizione delle diverse tariffe offerte, nel formato più oltre specificato.

## Parte 1

(punti: 20)

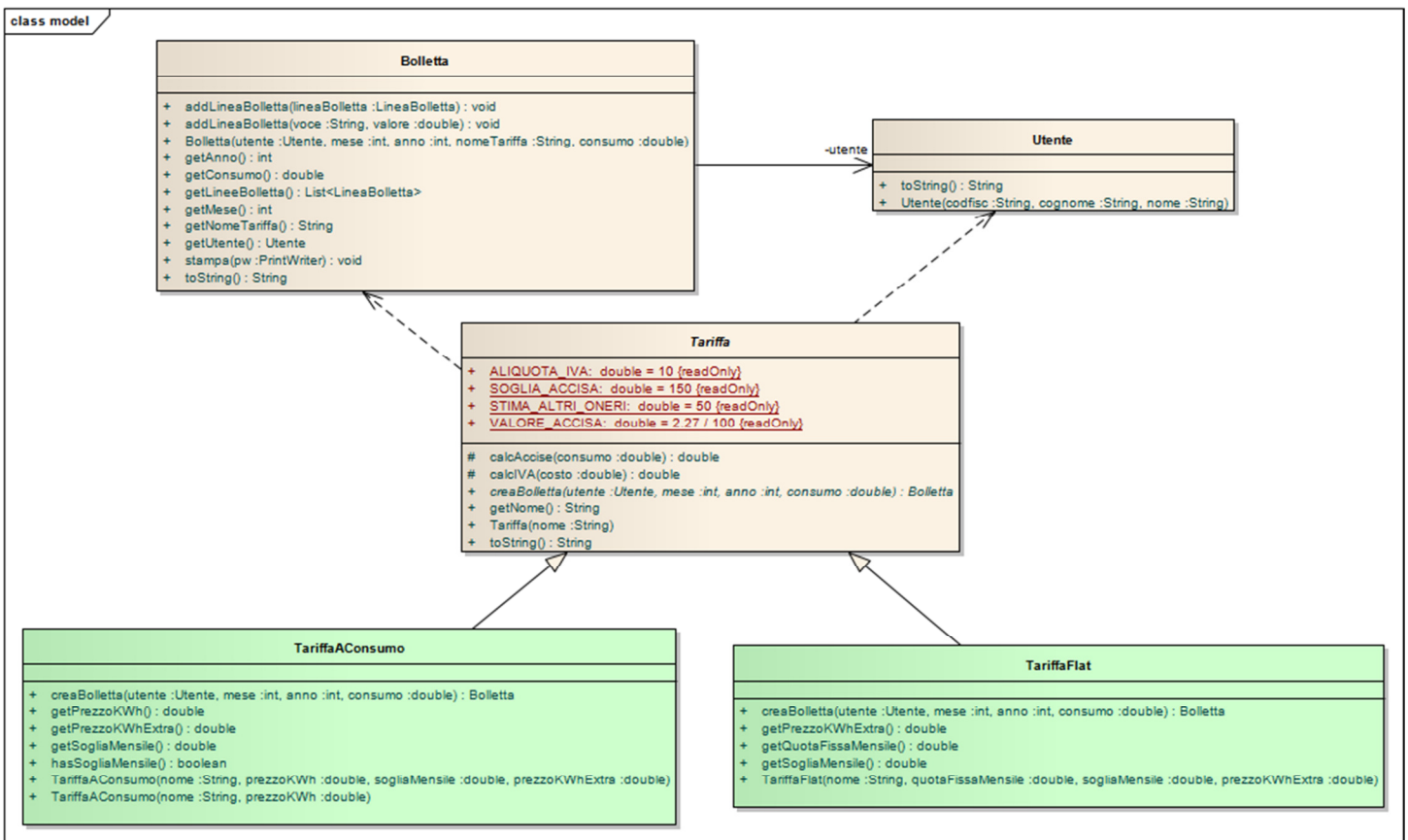
*Dati (namespace elettrodent.model)*

(punti: 13)

Il modello dei dati deve essere organizzato secondo il diagramma UML più sotto riportato.

SEMANTICA:

- la classe **Utente** (fornita nello start kit) rappresenta un utente, caratterizzato da codice fiscale, cognome e nome;
- la classe **LineaBolletta** (fornita nello start kit) rappresenta una voce di costo di una bolletta, caratterizzata dalle due proprietà *voce* e *valore*, ovviamente recuperabili tramite metodi accessor. Essa serve a descrivere la struttura della **Bolletta** (vedi oltre) in termini di lista di voci.



- c) la classe **Bolletta** (fornita nello start kit) rappresenta la bolletta di un dato utente in un dato mese e anno: essa è caratterizzata dal nome della tariffa applicata e dal consumo rilevato, nonché da una lista di **LineaBolletta**, ciascuna delle quali specifica una voce di costo presente in bolletta. La classe definisce quindi metodi per *aggiungere una linea di bolletta* (**addLineaBolletta**) e per recuperare la lista delle linee di bolletta presenti (**getLineeBolletta**), nonché una adeguata **toString** [oltre a un metodo accessorio **stampa** per produrre una stampa della bolletta su un apposito **PrintWriter** passato come argomento].
- d) la classe astratta **Tariffa** (fornita nello start kit) rappresenta una tariffa caratterizzata da un nome univoco, specificato all'atto della costruzione e recuperabile tramite opportuno metodo accessor; essa contiene inoltre la definizione di opportune costanti (**VALORE\_ACCISA**, **SOGLIA\_ACCISA**, **STIMA\_ALTRI\_ONERI**, **ALIQUOTA\_IVA**) per il calcolo delle varie voci. Il metodo **calcAccise** restituisce, dato un consumo in KWh, il valore in Euro delle corrispondenti accise, mentre il metodo **calcIVA** restituisce, dato un importo in Euro, il valore in Euro dell'IVA corrispondente. La classe dichiara infine il metodo astratto **creaBolletta** che, dato il consumo mensile, crea la corrispondente **Bolletta**, calcolando le voci di costo, le imposte e quindi l'importo totale come da specifiche.
- e) **la classe TariffaAConsumo (da realizzare)** specializza **Tariffa** nel caso a consumo ed è caratterizzata, oltre che dal nome, dai tre parametri prezzo base del KWh (in Euro), eventuale prezzo (in Euro) del KWh oltre una certa soglia di KWh mensili, eventuale valore di tale soglia; la classe deve prevedere due costruttori, uno a due argomenti (nome e prezzo base) e uno a quattro argomenti (nome, prezzo base, soglia, prezzo extra soglia). Tali valori sono forniti all'atto della costruzione e sono recuperabili tramite metodi accessor. La classe implementa **creaBolletta** inserendo nella bolletta le linee di bolletta opportune, e precisamente:
- 4 voci per le tariffe a consumo a prezzo singolo (senza soglia): *costo energia, accise, IVA, totale*;
  - 5 voci le tariffe a consumo a prezzi differenziati (con soglia): *costo energia entro soglia, costo energia extra soglia, accise, IVA, totale*
- f) **la classe TariffaFlat (da realizzare)** specializza **Tariffa** nel caso flat ed è caratterizzata, oltre che dal nome, dai tre parametri quota fissa mensile (in Euro), soglia mensile, prezzo degli eventuali KWh eccedenti la soglia (in Euro). Tali valori sono forniti all'atto della costruzione e sono recuperabili tramite metodi accessor. La classe implementa **creaBolletta** inserendo nella bolletta le linee di bolletta opportune, e precisamente:
- 5 voci: *quota fissa mensile, costo energia extra soglia, accise, IVA, totale*.

**Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di queste classi.**

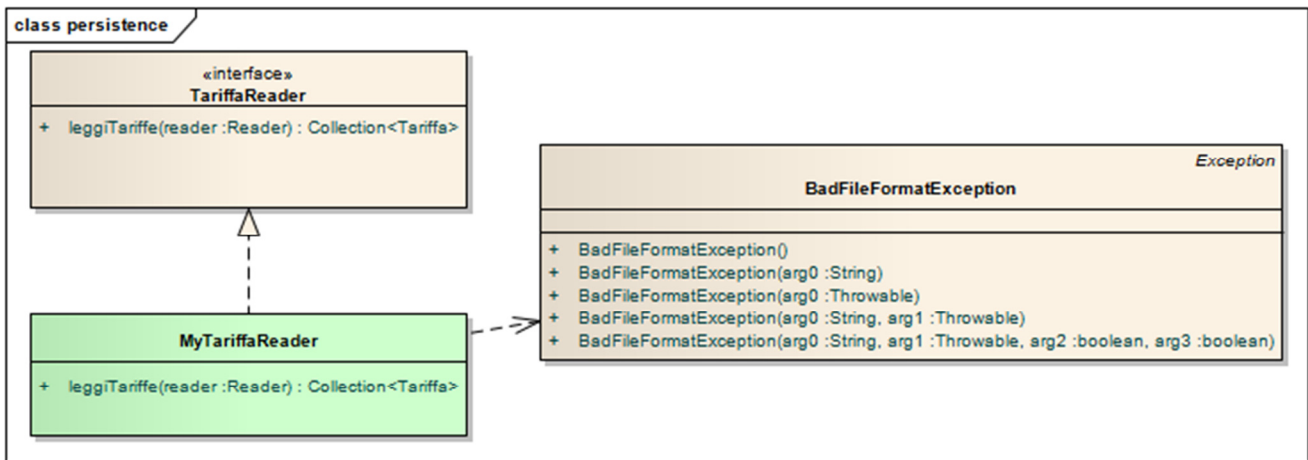
Come già anticipato, il file di testo `Tariffe.txt` contiene la descrizione delle diverse tariffe offerte, una per riga. Ogni riga contiene una serie di dati, separati fra loro da barre ('/'): gli elementi esatti della riga dipendono dal tipo di tariffa (a consumo o flat) e dalla eventuale presenza di alcuni dati opzionali. Più esattamente, ogni riga contiene:

- la tipologia della tariffa ("FLAT" o "A CONSUMO")
- il nome della tariffa (che può contenere spazi)
- poi:
  - per le tariffe flat: la soglia mensile (nel formato "SOGLIA" seguita da un intero che rappresenta i KWh inclusi), il prezzo mensile (nel formato "PREZZO" seguito da un valore in Euro), il costo dei KWh extra (nel formato "EXTRA" seguito da un valore in Eurocent);
  - per le tariffe a consumo: il prezzo del KWh (nel formato "PREZZO" seguito da un valore in Eurocent), opzionalmente seguito da un altro prezzo (sempre in Eurocent), dalla parola chiave "OLTRE" e da un intero che rappresenta la soglia di consumo oltre cui esso si applica (in KWh/mese);

```

ESEMPIO DEL FILE tariffe.txt
FLAT / CASA MINI      / SOGLIA 150 / PREZZO 29.00 / EXTRA 25
FLAT / CASA CLASSIC  / SOGLIA 250 / PREZZO 39.00 / EXTRA 25
FLAT / CASA BIG      / SOGLIA 350 / PREZZO 54.00 / EXTRA 25
FLAT / CASA MAXI    / SOGLIA 450 / PREZZO 74.00 / EXTRA 25
A CONSUMO / MONORARIA 1 / PREZZO 14.00
A CONSUMO / MONORARIA 2 / PREZZO 11.90 / 16.20 OLTRE 150
A CONSUMO / MONORARIA 3 / PREZZO 12.77 / 21.70 OLTRE 220
...
    
```

L'interfaccia `TariffaReader` (fornita) dichiara il metodo `leggiTariffe` che legge da un `Reader` una lista di `Tariffe`.

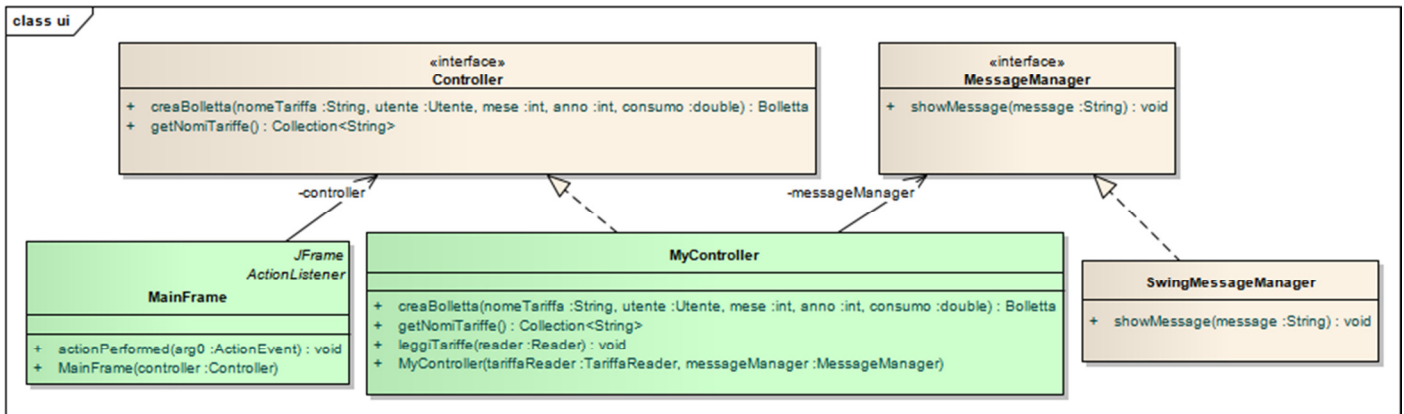


La classe `MyTariffaReader` (da realizzare) implementa tale interfaccia; oltre alla lettura del file, il metodo `leggiTariffe` deve effettuare i necessari controlli d'errore e lanciare l'eccezione `BadFileFormatException` (fornita nello start kit) quando opportuno.

*Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di questa classe.*

## Parte 2

La classe `MyController` (da realizzare) implementa l'interfaccia `Controller` (fornita): il costruttore riceve in ingresso un'istanza di una classe che implementa `TariffaReader` e un'istanza di una classe che implementa `MessageManager`. Quest'ultima interfaccia astrae un servizio di comunicazione con l'utente e contiene un metodo che consente di mostrare una *dialog* contenente un messaggio (es. un messaggio d'errore) – nello start kit è compresa l'implementazione da usare (`SwingMessageManager`). Il metodo `leggiTariffe` è un metodo pubblico di `MyController` che va usato nel main per predisporre il controller al funzionamento e che consente di caricare un insieme di tariffe

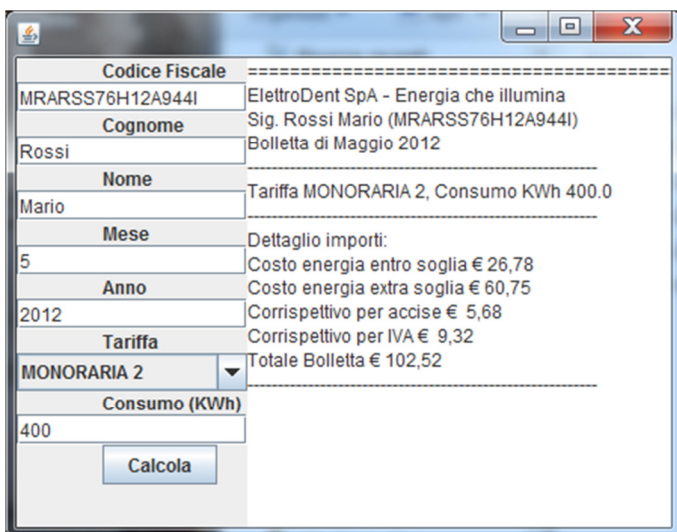


da un **Reader**; tale metodo deve usare **TariffaReader** per caricare tutte le tariffe e, dopo il caricamento, deve inserire le tariffe caricate in una mappa la cui chiave è il nome della tariffa e il cui valore è la tariffa stessa – il metodo deve catturare le eccezioni che si possono verificare durante il caricamento e deve comunicare all’utente il problema mediante l’uso del **MessageManager**. Inoltre, il metodo **getNomiTariffe** restituisce i nomi di tutte le tariffe contenute nella mappa mentre il metodo **creaBolletta** prende in ingresso tutti i dettagli necessari al calcolo di una bolletta (nome della tariffa, utente, mese, anno, consumo), recupera la tariffa dalla mappa mediante il nome, delega alla tariffa la creazione della **Bolletta** (metodo **creaBolletta** di **Tariffa**) e restituisce la **Bolletta** ottenuta.

**Lo Start Kit contiene anche i test (da includere nel progetto) per verificare il funzionamento di questa classe.**

*Interfaccia utente (namespace elettrodent.ui)*

*(punti 5)*



L’interfaccia utente deve essere simile (non necessariamente identica) all’esempio mostrato in figura.

La classe **Starter** (non mostrata nel diagramma UML, ma fornita nello start kit) contiene il main di partenza dell’intera applicazione.

**La classe MainFrame (da realizzare)** realizza la finestra principale, ed è così articolata:

- sulla sinistra, un pannello contenente un insieme di text field per l’input di un utente e dei relativi dati di consumo, oltre ad una combo box per la selezione della tariffa e ad un bottone (Calcola) per effettuare il calcolo;
- premendo il pulsante **Calcola**, viene mostrata a lato, nell’apposita *text area*, la bolletta corrispondente.

*Formato bolletta stampata*

```

=====
ElettroDent SpA - Energia che illumina
Sig. ROSSI MARIO (RSSMRA76H12A944I)
Bolletta di Maggio 2012
-----
Tariffa MONORARIA 3, Consumo KWh 200.0
-----
Dettaglio importi:
Costo energia entro soglia          € 38,31
Corrispettivo per accise            €  1,14
Corrispettivo per IVA               €  3,94
Totale Bolletta                    € 43,39
=====
  
```