

# ESAME DI FONDAMENTI DI INFORMATICA T-2 del 12/01/2016

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)**

È stata richiesta un'applicazione per confrontare i servizi offerti da agenzie *cambiavalute*. L'applicazione deve:

- preliminarmente, permettere di selezionare le due valute, **locale** ed **estera**, di interesse;
- far scegliere fra **acquisto** e **vendita** di valuta estera (vedere la spiegazione sottostante);
- consentire di *inserire l'importo da cambiare e mostrare l'importo cambiato corrispondente* sia secondo il tasso di cambio ufficiale sia secondo le offerte delle diverse agenzie (che adottano tassi di cambio diversi).

## DESCRIZIONE DEL DOMINIO DEL PROBLEMA

Ogni agenzia di cambio opera in un determinato Paese, avendo quindi come riferimento la *valuta locale* di quel Paese. L'operazione di cambio valuta può consistere in:

- **vendita** al cliente di una certa quantità di valuta estera (pagata dal cliente in valuta locale);
- **acquisto** dal cliente di una certa quantità di valuta estera (dandogli in cambio valuta locale).

Ad esempio, per un'agenzia di cambio italiana, che ha come valuta locale di riferimento l'Euro, un cambio di *Dollari in Euro* costituisce un *acquisto* di valuta estera (i dollari), mentre un cambio di *Euro in Sterline* è una *vendita* (di sterline). La stessa operazione fatta in un altro Paese può implicare l'inversione dei concetti di acquisto e vendita, perché cambia ciò che si considera "locale" ed "estero": ad esempio, il cambio di Dollari in Euro citato sopra, se fatto negli USA, sarebbe visto come *vendita* di Euro, mentre il cambio di Euro in Sterline fatto in Gran Bretagna sarebbe un *acquisto* di Euro.

*Il guadagno del cambiavalute si gioca sui prezzi di acquisto e vendita*, che sono rispettivamente **più alto/più basso** del tasso ufficiale di cambio (perché definiti rispetto a una unità di valuta locale). Di conseguenza, quando il cliente chiede valuta estera (vendita da parte dell'agenzia al cliente) ne riceve un po' meno di quanto dovrebbe secondo il tasso ufficiale, mentre quando cede valuta estera (acquisto da parte dell'agenzia), riceve un po' meno valuta locale di quanto dovrebbe. Questo margine può variare molto da un'agenzia a un'altra, come mostrato negli esempi sottostanti.

Ogni valuta è identificata internazionalmente da una *sigla di tre lettere* di cui in genere le prime due indicano il paese, mentre la terza è l'iniziale del nome della valuta: ad esempio, USD è il dollaro americano, CHF il franco svizzero, GBP la sterlina (pound) britannica; fa eccezione l'euro, identificato semplicemente dalla sigla EUR.

### Esempio 1 - Cambio di 100 Euro in Dollari USA in Italia (valuta locale: EUR)

- cambio ufficiale:  $1.00 \text{ EUR} = 1.0947 \text{ USD} \rightarrow 100.00 \text{ EUR} = 109.47 \text{ USD}$  (teorico)
- prezzo di vendita presso Agenzia1:  $1.00 \text{ EUR} = 1.0162 \text{ USD} \rightarrow 100.00 \text{ EUR} = 101.62 \text{ USD}$  ☹
- prezzo di vendita presso Agenzia2:  $1.00 \text{ EUR} = 1.0720 \text{ USD} \rightarrow 100.00 \text{ EUR} = 107.20 \text{ USD}$  😊

### Esempio 2 - Cambio di 200 Dollari USA in Euro in Italia (valuta locale: EUR)

- cambio ufficiale:  $1.00 \text{ EUR} = 1.0947 \text{ USD} \rightarrow 1.00 \text{ USD} = 0.9135 \text{ EUR} \rightarrow 200 \text{ USD} = 182.70 \text{ EUR}$
- prezzo di acquisto Agenzia1:  $1.00 \text{ EUR} = 1.2314 \text{ USD} \rightarrow 1.00 \text{ USD} = 0.8121 \text{ EUR} \rightarrow 200 \text{ USD} = 162.42 \text{ EUR}$  ☹
- prezzo di acquisto Agenzia2:  $1.00 \text{ EUR} = 1.1157 \text{ USD} \rightarrow 1.00 \text{ USD} = 0.8963 \text{ EUR} \rightarrow 200 \text{ USD} = 179.26 \text{ EUR}$  😊

### Esempio 3 - Cambio di 100 Euro in Franchi Svizzeri presso banca a Lugano (valuta locale: CHF)

- cambio ufficiale:  $1.00 \text{ CHF} = 0.9254 \text{ EUR} \rightarrow 1.00 \text{ EUR} = 1.0806 \text{ CHF} \rightarrow 100.00 \text{ EUR} = 108.06 \text{ CHF}$
- prezzo di acquisto:  $1.00 \text{ CHF} = 0.9443 \text{ EUR} \rightarrow 1.00 \text{ EUR} = 1.0590 \text{ CHF} \rightarrow 100.00 \text{ EUR} = 105.90 \text{ CHF}$

### Esempio 4 - Cambio di 100 sterline in Euro presso agenzia con sede in Gran Bretagna (valuta locale: GBP)

- cambio ufficiale:  $1.00 \text{ GBP} = 1.3620 \text{ EUR} \rightarrow 100.00 \text{ GBP} = 136.20 \text{ EUR}$
- prezzo di vendita:  $1.00 \text{ GBP} = 1.3296 \text{ EUR} \rightarrow 100.00 \text{ GBP} = 132.96 \text{ EUR}$

Vari file di testo, di nome *nomeagenzia.txt*, contengono ciascuno i tassi di cambio offerti da una specifica banca o agenzia, mentre il file di testo *CambiUfficiali.csv* contiene i tassi di cambio ufficiali di *tutte le valute*.

## Parte 1

(punti: 18)

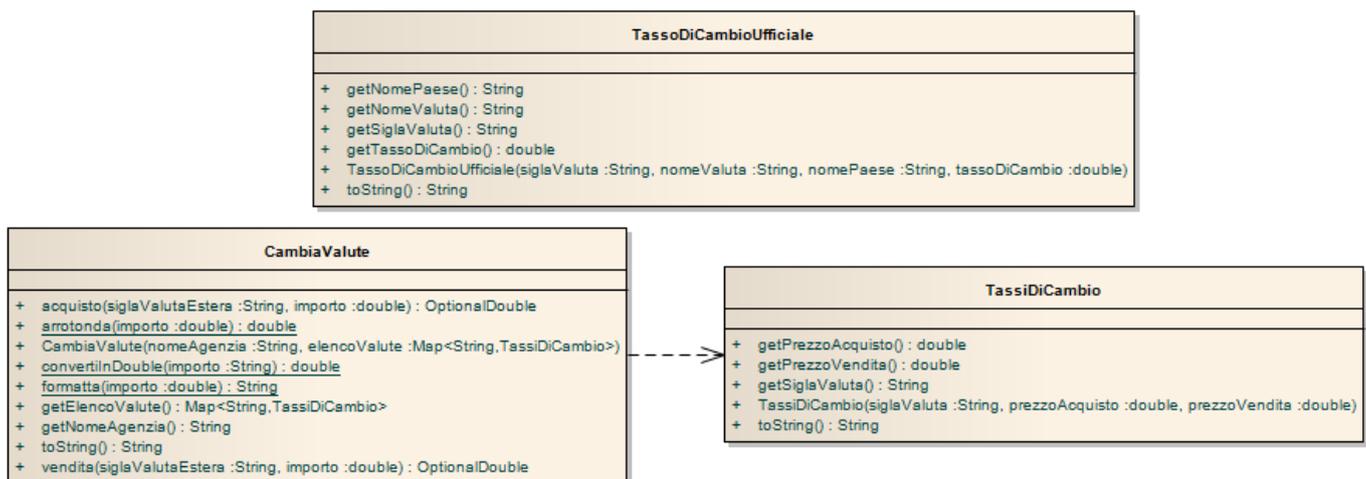
Dati (namespace *cambiavalute.model*)

(punti: 8)

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

SEMANTICA:

- la classe **TassoDiCambioUfficiale** (fornita nello start kit) rappresenta il tasso di cambio ufficiale di una data valuta estera, definita dalle seguenti proprietà: sigla internazionale della valuta (ad esempio, “USD”), nome esteso della valuta (ad esempio, “Dollaro USA”), nome del Paese corrispondente (ad esempio, “STATI UNITI”) e tasso di cambio, che esprime il controvalore di 1 Euro in valuta estera (ad esempio, 1.0887 USD); sono disponibili gli accessor a queste proprietà nonché una opportuna **toString**. Ovviamente il costruttore riceve i medesimi dati.
- la classe **TassiDiCambio** (fornita nello start kit) rappresenta i due tassi di cambio (acquisto e vendita) di una data valuta estera, che saranno usati in una qualche agenzia cambiavalute; le proprietà sono semplicemente la sigla internazionale della valuta (ad esempio, “USD”) e i due prezzi, ciascuno dei quali esprime il controvalore di 1 Euro rispettivamente in caso di acquisto/vendita di valuta estera (ad esempio, 1.1290/1.0596 USD). Sono disponibili gli accessor e una opportuna **toString**. Anche qui, il costruttore riceve i medesimi dati.
- la classe **CambiaValute (da realizzare)** rappresenta un’agenzia cambiavalute, caratterizzata da un nome e dalla tabella di **TassiDiCambio** che utilizza per le sue operazioni di compravendita. Questi dati sono entrambi forniti al costruttore: in particolare, la tabella è una **Map<String,TassiDiCambio>**, dove la chiave è la sigla della valuta (ad esempio, “USD”) e il valore associato è l’opportuna istanza di **TassiDiCambio**. La classe deve fornire i metodi:
  - getNomeAgenzia** e **getElencoValute**, gli accessor alle due proprietà ricevute dal costruttore;
  - toString** che restituisce semplicemente il nome dell’agenzia cambiavalute;
  - vendita(String siglaValutaEstera, double importo)** che calcola l’equivalente in valuta **estera** dell’importo passato (che si considera in valuta **locale**): poiché l’agenzia potrebbe non gestire quella valuta, il risultato è un **OptionalDouble** che incapsula il risultato se esso è presente, o è *empty* in caso contrario.
  - acquisto(String siglaValutaEstera, double importo)** che calcola l’equivalente in valuta **locale** dell’importo passato (che si considera in valuta **estera**): poiché l’agenzia potrebbe non gestire quella valuta, il risultato è un **OptionalDouble** che incapsula il risultato se esso è presente, o è *empty* in caso contrario.



Deve inoltre fornire i seguenti metodi statici di utilità:

- arrotonda** che, dato un valore double, lo restituisce arrotondato a due cifre decimali;
- formatta** che, dato un valore double, restituisce una stringa formattata con due cifre decimali secondo il **Locale.ITALY**, in modo da utilizzare la virgola anziché il punto decimale;
- convertiInDouble** che, data una stringa formattata con due cifre decimali secondo il **Locale.ITALY**, restituisce il valore double corrispondente, lanciando **ParseException** in caso di stringa inammissibile. A

questo proposito, si tenga presente che **NumberFormat.parse** impostato con il locale italiano considera il punto decimale come separatore delle migliaia, cosa che può facilmente confondersi con una virgola (separatore dei decimali). Perciò, in caso di presenza del punto decimale nella stringa in ingresso è esplicitamente richiesto di intercettare la situazione e lanciare una **ParseException**.

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

**Persistenza (namespace *cambiavalute.persistence*)**

(punti 10)

Come già anticipato, vari file di testo, di nome *nomeagenzia.txt*, contengono ciascuno i tassi di cambio offerti da una data banca o agenzia cambiavalute. **È compito del main nella classe Program (fornita) recuperare l'elenco di tali file e invocare su ciascuno di essi l'apposito reader**, che quindi deve preoccuparsi soltanto di leggere *un singolo file*.

Le righe nel file contengono i tassi di cambio delle valute gestite in quell'agenzia, una valuta per riga. Più precisamente, ogni riga contiene tre elementi, separati da spazi e/o tabulazioni:

- la sigla internazionale della valuta estera (tre lettere)
- il prezzo di vendita (un numero reale **che utilizza la virgola come separatore della parte decimale**)
- il prezzo di acquisto (un numero reale **che utilizza la virgola come separatore della parte decimale**)

È importante sottolineare che:

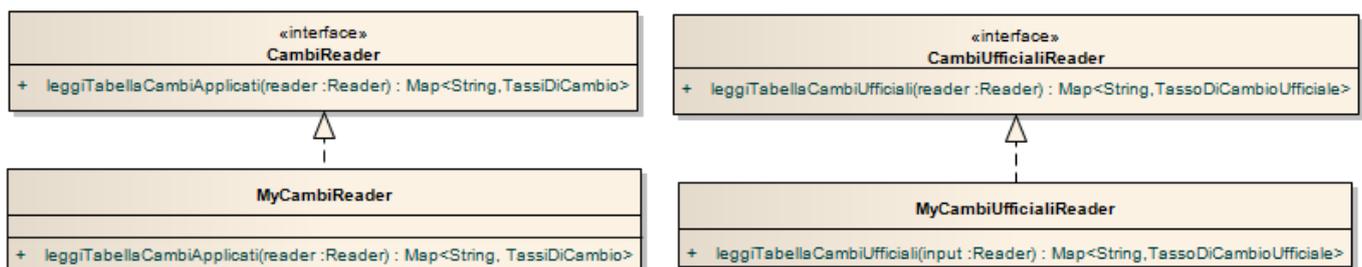
- ogni agenzia è libera di trattare certe valute e non altre: pertanto, i file non hanno un numero di righe fisso;
- le valute trattate sono elencate in ordine qualsiasi (in particolare, non alfabetico).

Il file di testo *CambiUfficiali.csv* contiene invece i tassi di cambio ufficiali rilevati dalla Banca d'Italia: ogni riga contiene il cambio di una valuta, insieme a molte informazioni accessorie, tutte separate da virgole. Nell'ordine troviamo:

- il nome del Paese estero
- il nome della valuta
- la sigla internazionale della valuta
- il codice internazionale del Paese estero (*irrilevante nel nostro caso*)
- il tasso di cambio ufficiale (un numero reale che adotta il punto come separatore della parte decimale)
- molte altre informazioni accessorie, irrilevanti in questa sede.

<pre>ESEMPIO EDbank.txt AUD 1,4411 1,5863 JPY 128,2541 135,9345 ... ESEMPIO ZannoChange.txt USD 1,0720 1,1157 JPY 129,0807 134,3493 ...</pre>	<pre>ESEMPIO CambiUfficiali.csv CANADA,Dollaro Canadese,CAD,012,1.5116,... REGNO UNITO,Sterlina Gran Bretagna,GBP,002,0.73395,... COREA DEL SUD,Won Sud,KRW,119,1280.78,... STATI UNITI,Dollaro USA,USD,001,1.0887,... DANIMARCA,Corona Danese,DKK,007,7.4626,... SVIZZERA,Franco Svizzero,CHF,003,1.0835,...</pre>
-----------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Sono fornite le due interfacce **CambiReader** e **CambiUfficialiReader** che dichiarano ciascuna un singolo metodo di lettura, di nome rispettivamente **leggiTabellaCambiApplicati** e **leggiTabellaCambiUfficiali**: entrambi ricevono come argomento un **Reader** e restituiscono un'opportuna mappa avente per chiave la sigla della valuta e per valore associato rispettivamente l'opportuna istanza di **TassiDiCambio** o di **TassiDiCambioUfficiale**.



Le due classi **MyCambiReader** e **MyCambiUfficialiReader** implementano tali interfacce effettuando i necessari controlli sul formato del file e lanciando **BadFileFormatException** (fornita) quando opportuno. Non sono richiesti costruttori poiché nessuno dei due reader ha necessità di mantenere uno stato interno esplicito.

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

## Parte 2

(punti: 12)

### Controller (cambiavalute.ui.controller)

(punti 4)

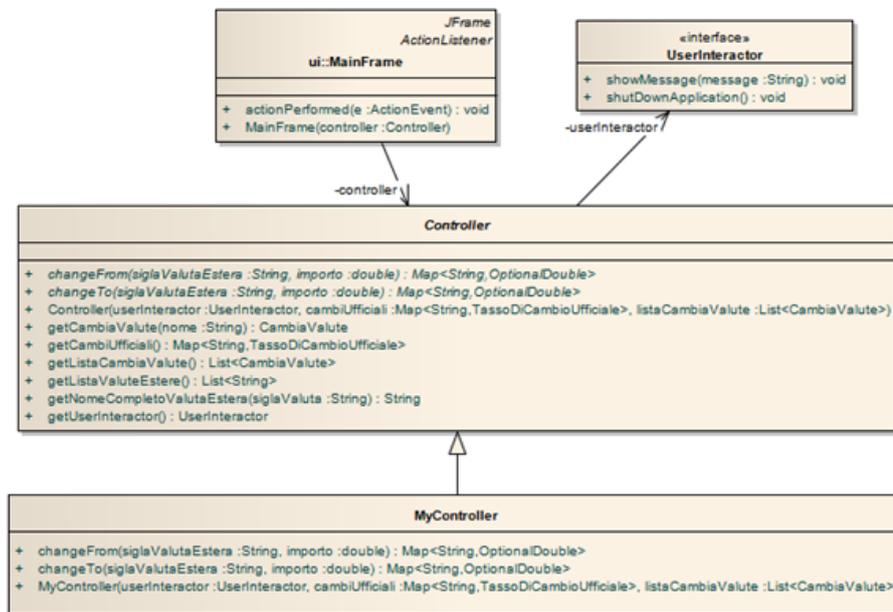
La classe astratta **Controller** (fornita nello start kit) costruisce un controller a partire da tre argomenti – uno **UserInteractor** (fornito) per gestire i messaggi d’errore, la mappa dei cambi ufficiali (ottenuta dal reader) e la lista dei **CambiaValute** disponibili (fornita dal main in **Program**). Il controller offre già implementati i seguenti metodi:

- **getUserInteractor**, **getCambiUfficiali** e **getListaCambiaValute** (gli accessor alle tre proprietà);
- **getListaValuteEstere** che restituisce la lista delle sigle di tutte le valute estere censite nei cambi ufficiali;
- **getCambiaValute(String nomeAgenzia)** che restituisce l’istanza di **CambiaValute** avente il nome specificato;
- **getNomeCompletoValutaEstera(String siglaValuta)** che restituisce il nome completo (come letto dal file dei cambi ufficiali) di una data valuta estera di cui sia nota la sigla.

Dichiara inoltre i due metodi astratti:

- **changeTo** e **changeFrom** che, data una valuta estera e un importo, restituiscono una tabella (sotto forma di **Map<String,OptionalDouble>**) contenente i risultati dei cambi verso/da quella valuta estera effettuati da tutti i cambiavalute (ovviamente se un dato cambiavalute non tratta quella valuta il relativo risultato sarà *empty*);

La classe **MyController** concretizza **Controller** implementando i due metodi astratti; il costruttore rimappa semplicemente il lavoro sul costruttore della classe base.



### Interfaccia Utente (cambiavalute.ui)

(punti 8)

L’interfaccia utente deve essere simile (non necessariamente identica) agli esempi mostrati nelle figure seguenti.

La classe **Program** (non mostrata nel diagramma UML illustrato alla pagina seguente, ma fornita nello start kit) contiene il main di partenza dell’intera applicazione; come già anticipato, è il main che si occupa di recuperare l’elenco dei file **nomeagenzia.txt** e aprire su ciascuno di essi l’apposito reader.

La classe **MainFrame (da realizzare)** costituisce la finestra principale dell’applicazione. Organizzativamente:

- la combo **Valuta** è pre-popolata con tutte le valute estere di cui si conosca il cambio ufficiale;

- il gruppo di radiobutton è inizialmente pre-settato su *Vendita*;
- l'importo da cambiare è inizialmente pre-settato su 100,00 unità di valuta (locale o estera, dipende se si tratta di vendita o acquisto, rispettivamente)
- l'importo cambiato al cambio ufficiale (non editabile) è inizialmente vuoto;
- la textarea di uscita (anch'essa non editabile) è inizialmente vuota.

Tutti gli importi devono essere formattati secondo il Locale.Italy con esattamente due cifre decimali.

**L'applicazione deve reagire a qualunque modifica fatta su uno qualunque dei componenti attivi** (la combo, i radiobutton, l'importo da cambiare), ricalcolando l'operazione in base ai nuovi dati presenti nella GUI.

Se nel campo importo viene inserito un valore non numerico o non riconosciuto come tale (ad esempio perché non è un numero o è presente il punto decimale invece della virgola) deve essere emesso un messaggio d'errore tramite lo **UserInteractor**, svuotando al contempo i due campi di uscita (v. ultime due figure).

