

ESAME DI FONDAMENTI DI INFORMATICA T-2 dell' 8/02/2016

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 e CARTELLA : CognomeNome-matricola (es. RossiMario-0000123456)

NOME ZIP DA CONSEGNARE : CognomeNome-matricola.zip (es. RossiMario-0000123456.zip)

Il Ministero delle Finanze della Repubblica di Dentinia ha richiesto di sviluppare un'applicazione che agevoli i cittadini nella compilazione della dichiarazione dei redditi e permetta di calcolare il dovuto. L'applicazione deve:

- permettere di inserire i propri **redditi** e le proprie **spese**;
- consentire di scegliere la propria **tipologia** di contribuente, a cui corrisponde una diversa **no-tax area**;
- mostrare l'**imposta lorda** con la corrispondente **aliquota media**, nonché l'**imposta netta** finale ottenuta scontando dall'imposta lorda le **detrazioni per spese mediche**.

DESCRIZIONE DEL DOMINIO DEL PROBLEMA

La tassazione delle persone fisiche nella Repubblica di Dentinia è progressiva, ossia si applicano aliquote percentuali crescenti al crescere del reddito del contribuente. A tal fine, i redditi sono divisi in fasce:

- **fino a 15.000 euro**: aliquota del 10%;
- **oltre 15.000 euro e fino a 25.000 euro**: aliquota del 25%;
- **oltre 25.000 euro e fino a 45.000 euro**: aliquota del 33%;
- **oltre 45.000 euro e fino a 70.000 euro**: aliquota del 38%;
- **oltre 70.000 euro**: aliquota del 45%.

Per attenuare l'impatto sulle categorie più deboli è prevista però una **no-tax area**, ossia una quota di reddito non soggetta da alcuna tassazione, la cui ampiezza a dipende dalla **tipologia** del contribuente. Più precisamente:

- **per lavoratori dipendenti e studenti**: no-tax area = 8.000 euro
- **per lavoratori autonomi**: no-tax area = 5.000 euro

Per questo il **reddito imponibile** (effettivamente soggetto a tassazione) è minore di quello **lordo** complessivo.

Ad esempio, un lavoratore *dipendente* con un reddito *lordo* di 40.000 euro avrà un reddito *imponibile* di 32.000 euro, mentre un lavoratore *autonomo* di pari reddito lordo avrà un *imponibile* di 35.000 euro.

Su questi valori si applica poi il sistema progressivo sopra descritto, partendo dalla fascia più alta applicabile e passando via via alla precedente per la quota-parte di reddito che rimane da tassare.

Esempio - *imponibile di 32.000 euro* → 3° scaglione (25.000-45.000 euro)

- quota parte nel 3° scaglione: $32.000 - 25.000 = 7.000$ euro → aliquota del 33% → imposta = 2.310 euro
- quota parte nel 2° scaglione: $25.000 - 15.000 = 10.000$ euro → aliquota del 25% → imposta = 2.500 euro
- quota parte nel 1° scaglione: $15.000 - 00.000 = 15.000$ euro → aliquota del 10% → imposta = 1.500 euro
- imposta complessiva: $2.310 + 2.500 + 1.500 = 6.310$ euro

L'imposta complessiva così ottenuta si dice **imposta lorda** perché non è ancora il valore realmente da pagare: la Repubblica di Dentinia permette infatti di scontare una parte – il 20% - delle **spese mediche** sostenute. Il risultato costituisce l'**imposta netta** effettivamente da pagare.

Ad esempio, a fronte di spese mediche per 1.400 euro potranno essere scontati dall'imposta *lorda* ulteriori 280 euro, giungendo così a una imposta *netta* – nel caso dell'esempio sopra – di 6.030 euro.

L'imposta lorda costituisce però la base di calcolo della cosiddetta **aliquota media**, definita come rapporto fra l'imposta lorda e il reddito del contribuente - nel caso dell'esempio sopra, $6.310 / 40.000 = 15,78\%$.

Il file di testo [Scaglioni.txt](#) contiene gli scaglioni progressivi di reddito con le relative aliquote, mentre il file di testo [No-tax area.txt](#) specifica l'ampiezza della no-tax area per le varie tipologie di contribuenti considerate.

Parte 1

(punti: 20)

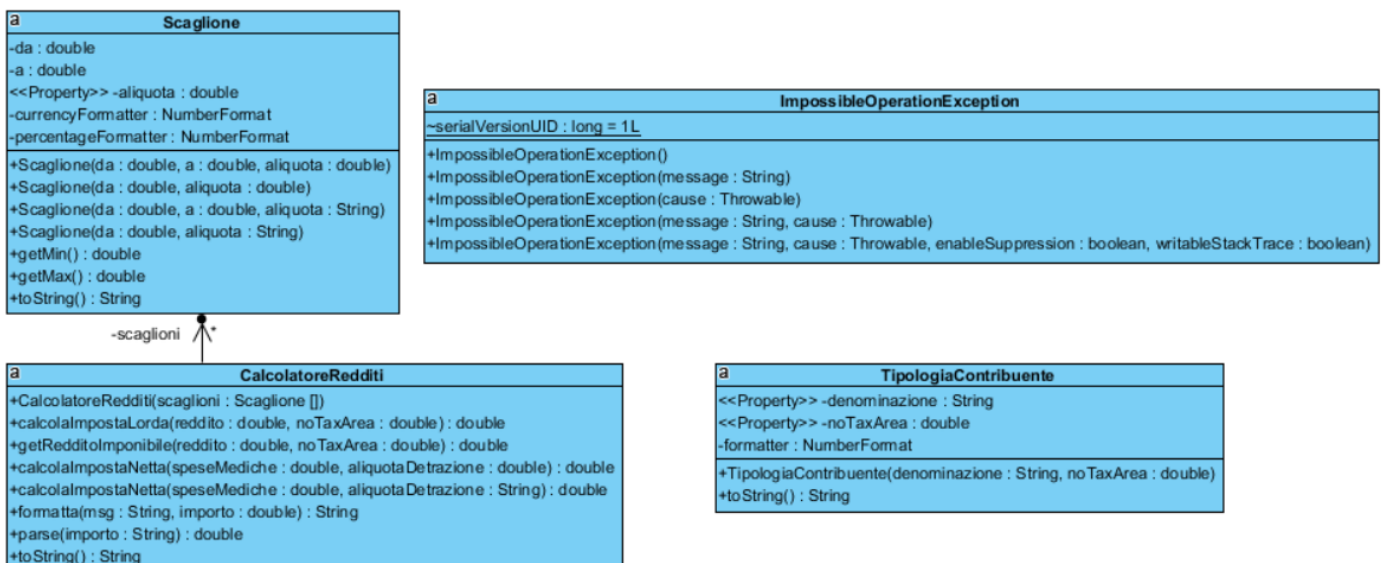
Dati (namespace redditi.model)

(punti: 12)

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

SEMANTICA:

- la classe **Scaglione** (fornita nello start kit) rappresenta uno scaglione reddituale, definito dalle tre proprietà: importo inferiore (incluso), importo superiore (escluso), aliquota applicabile; sono disponibili vari costruttori e gli accessor a queste proprietà nonché una opportuna **toString**.
- la classe **TipologiaContribuente** (fornita nello start kit) rappresenta un tipo di contribuente con la relativa no-tax area; sono disponibili un costruttore e gli accessor a queste proprietà nonché una opportuna **toString**.
- la classe **CalcolatoreRedditi** (da realizzare) effettua il calcolo delle imposte come da algoritmo specificato nel dominio del problema. Il costruttore riceve un array di **Scaglione**, dove l'indice 0 rappresenta lo scaglione inferiore. Il calcolatore mantiene nel suo stato interno l'imposta lorda e netta relative all'ultimo calcolo effettuato, sotto forma di due oggetti **OptionalDouble**. La classe deve fornire i seguenti metodi:
 - getRedditoImponibile(double reddito, double noTaxArea)** restituisce la differenza fra il reddito dato e la soglia di no-tax area specificata, o 0 se la no-tax area assorbe interamente il reddito;
 - getScaglioni** restituisce l'array degli scaglioni di reddito, ricevuto dal costruttore;
 - toString** restituisce semplicemente l'elenco degli scaglioni di reddito, uno per riga;
 - calcolaImpostaLorda(double reddito, double noTaxArea)** calcola l'imposta lorda corrispondente al reddito dato, tenendo conto della soglia di no-tax area specificata; memorizza tale valore nello stato interno e lo restituisce;



- calcolaImpostaNetta(double speseMediche, double aliquotaDetrazione)** calcola l'imposta netta a partire dalla imposta lorda memorizzata nello stato interno del calcolatore, applicando la detrazione per le spese mediche specificate con l'aliquota percentuale specificata; se l'imposta lorda non è stata ancora calcolata, lancia **ImpossibleOperationException** (fornita); altrimenti memorizza il risultato nello stato interno e lo restituisce;
- calcolaImpostaNetta(double speseMediche, String aliquotaDetrazione)** idem come sopra, ma il secondo argomento è una stringa che esprime una percentuale (ad esempio, "20%"): il metodo effettua il parsing di tale stringa (lanciando **java.text.ParseException** se essa non rappresenta una percentuale) e richiama il metodo omonimo sopra descritto per proseguire il calcolo;

- **formatta(String msg, double importo)** formatta l'importo come valuta, tramite un opportuno CurrencyFormatter, e restituisce la stringa ottenuta anteponendo ad esso il messaggio *msg*;
- **parse(String importo)** effettua il parsing dell'importo ricevuto (lanciando **java.text.ParseException** se l'importo non è un numero correttamente formattato) e restituisce il valore double corrispondente;

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

Persistenza (namespace redditi.persistence)

(punti 8)

Come già anticipato, il file di testo **Scaglioni.txt** contiene gli scaglioni progressivi di reddito con le relative aliquote, mentre il file di testo **No-tax area.txt** specifica l'ampiezza della no-tax area per le varie tipologie di contribuenti.

Il file **Scaglioni.txt** contiene i vari scaglioni reddituali, uno per riga. Ogni riga ha il seguente formato:

LimiteInferiore-LimiteSuperiore: aliquotaPercentuale

dove:

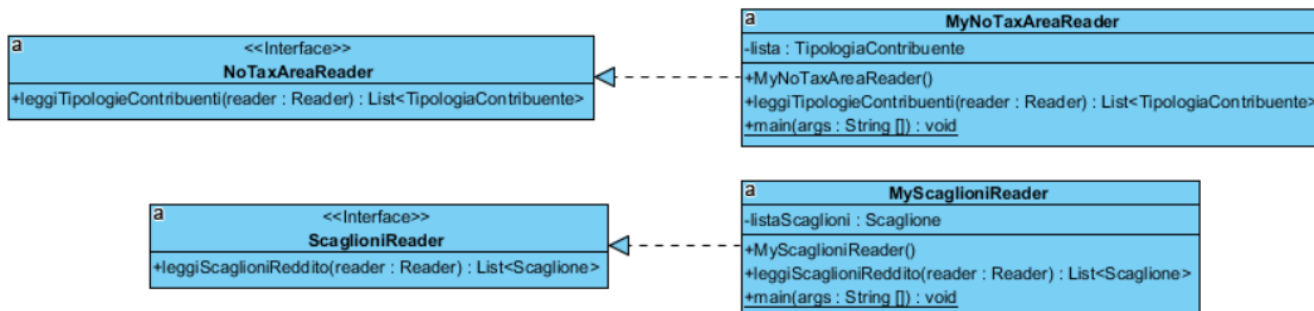
- *LimiteInferiore* e *LimiteSuperiore* sono due valori numerici formattati "all'italiana", ossia con il punto come separatore delle migliaia;
- *aliquotaPercentuale* è una percentuale
- il carattere '-' separa i due limiti fra loro
- il carattere ':' separa il limite superiore dalla percentuale

E' garantito che gli scaglioni siano indicati dal minore al maggiore, e che ciascuno di essi abbia come limite superiore il limite inferiore dello scaglione successivo.

Il file di testo **No-tax area.txt** specifica l'ampiezza della no-tax area per le varie tipologie di contribuenti, una per riga. Ogni riga contiene quindi la descrizione della tipologia (che può contenere spazi) e l'importo della no-tax area corrispondente, separati dal carattere ':'

<p><i>ESEMPIO Scaglioni.txt</i></p> <pre>0-15.000: 10% 15.000-25.000: 25% 25.000-45.000: 33% 45.000-70.000: 38% 70.000-oltre: 45% ...</pre>	<p><i>ESEMPIO No-tax area.txt</i></p> <pre>lavoratori dipendenti: 8.000 lavoratori autonomi: 5.000 ...</pre>
---	--

Sono fornite le due interfacce **ScaglioniReader** e **NoTaxAreaReader** che dichiarano ciascuna un singolo metodo di lettura, di nome rispettivamente **leggiScaglioniReddito** e **leggiTipologieContribuenti**: entrambi ricevono come argomento un **Reader** e restituiscono la corrispondente lista di **Scaglione** o di **TipologiaContribuente**.



Le due classi **MyScaglioniReader** e **MyNoTaxAreaReader** 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: 10)

Controller (redditi.ui.controller)

(punti 3)

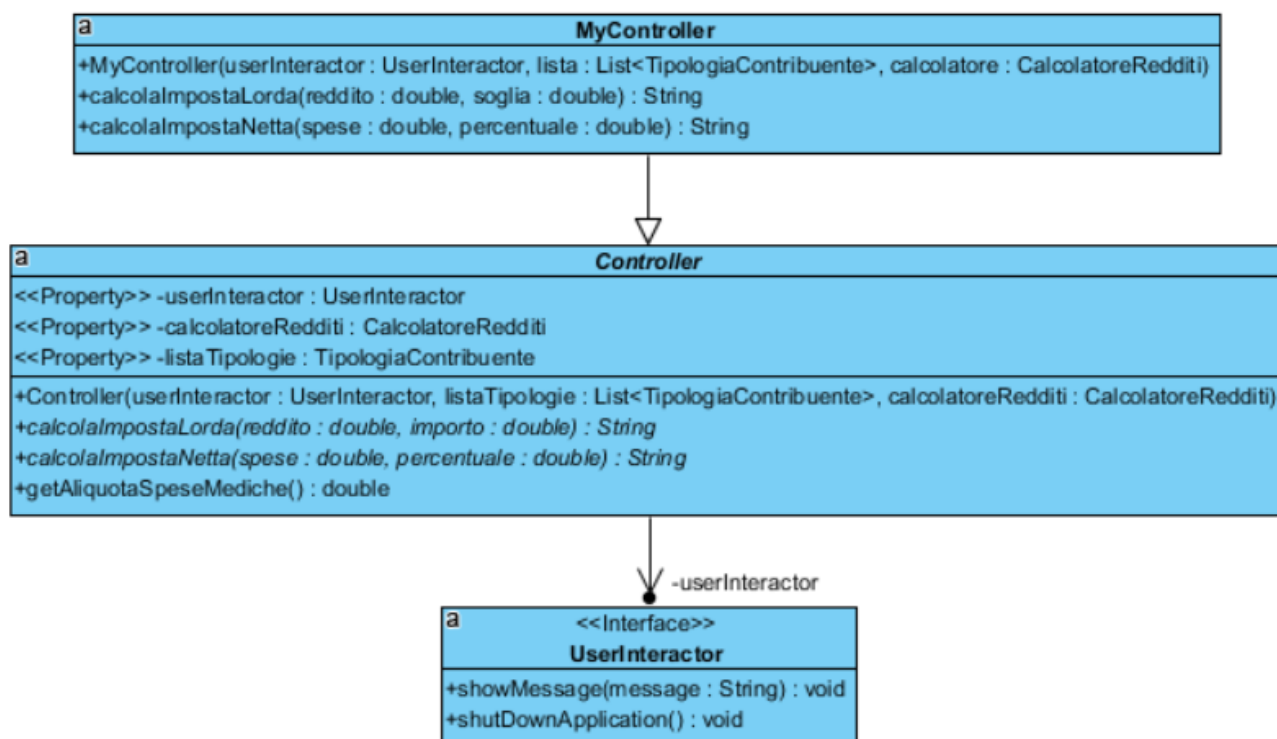
La classe astratta **Controller** (fornita nello start kit) costruisce un controller a partire da tre argomenti – uno **UserInteractor** (fornito) per gestire i messaggi d'errore, la lista delle tipologie contribuenti e il **CalcolatoreRedditi** prescelto (fornito dal main in **Program**). Il controller offre già implementati i seguenti metodi:

- **getUserInteractor**, **getListaTipologie** e **getCalcolatoreRedditi** (gli accessor alle tre proprietà);
- **getAliquotaSpeseMediche** che restituisce l'aliquota applicabile per le detrazioni delle spese mediche, fissa al 20% (0.2);

Dichiara inoltre i due metodi astratti:

- **calcolaImpostaLorda(double reddito, double importo)** e **calcolaImpostaNetta(double spese, double percentuale)** che restituiscono una stringa formattata come valuta contenente l'importo dell'imposta lorda o netta, rispettivamente.

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



Interfaccia Utente (redditi.ui)

(punti 7)

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 **Tipologia** è pre-popolata con tutte le tipologie di contribuenti (e relative no-tax area);
- il reddito lordo e le spese mediche sono inizialmente pre-settati su 0,00;
- il reddito imponibile (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 due campi di testo editabili), ricalcolando le imposta lorda e netta in base ai nuovi dati presenti nella GUI.

Se nel campo importo viene inserito un valore non numerico deve essere emesso un messaggio d'errore tramite lo **UserInteractor**, svuotando al contempo i due campi di uscita (v. ultima figura).

Dentinia Redditi

Tipologia: lavoratori dipendenti € 8.000,00

Reddito lordo: 0,00

Imponibile:

Spese mediche: 0,00

Dentinia Redditi

Tipologia: lavoratori dipendenti € 8.000,00

Reddito lordo: 0,00

Imponibile:

Spese mediche: 0,00

Dentinia Redditi

Tipologia: lavoratori autonomi € 5.000,00

Reddito lordo: 40000,00

Imponibile: € 35.000,00

Spese mediche: 0,00

Imposta lorda € 7.300,00
imposta netta € 7.300,00

Dentinia Redditi

Tipologia: lavoratori dipendenti € 8.000,00

Reddito lordo: 0,00

Imponibile:

Spese mediche: 0,00

Dentinia Redditi

Tipologia: lavoratori dipendenti € 8.000,00

Reddito lordo: pippo

Imponibile:

Spese mediche: 0,00

Messaggio: Gli importi devono essere numerici

OK