



Istituto Tecnico Tecnologico Statale "Alessandro Volta"

Scuol@2.0

Chimica, Materiali e Biotecnologie
Grafica e Comunicazione
Trasporti e Logistica

Via Assisana, 40/E - loc. Piscille - 06135 Perugia
Centralino 075.31045 fax 075.31046 C.F. 80005450541
www.avolta.pg.it
voltauffici@tin.it dirigente@avolta.pg.it

Meccanica, Meccatronica ed Energia
Elettronica ed Elettrotecnica
Informatica e Telecomunicazioni

Corso di
Tecnologie e progettazione di sistemi informatici e di
telecomunicazioni

Classi 5AInf, 5BInf, 5CInf
a.s. 2016/2017

Relazione progetto
"Titolo del progetto"

Studente: Nicolò Vescera

Docente/i: Monica Ciuchetti

Documento di Specifica dei Requisiti (Software Requirements Specification)

1. Introduzione e formulazione del problema

Questo programma in Java simula la gestione di un Contocorrente con vari utenti, chiamati Correntisti, che possono andare ad operare con 2 azioni: Versamento e Prelievo, sui soldi presenti nel conto. Un solo Correntista alla volta può accedere al conto e gli altri devono aspettare che il Contocorrente si liberi prima di poter eseguire qualunque azione. Ogni azione di ogni correntista viene registrata nel Contocorrente così da poter vedere quale correntista ha eseguito una determinata azione. Quando tutti i correntisti hanno finito le loro azioni verrà stampata a video l'intera lista movimenti dei correntisti e il saldo finale del conto.

2. Descrizione dell'architettura dell'applicazione (componenti hardware e software)

Questo programma è stato scritto e compilato in un ambiente linux (Ubuntu 16.10), con la versione 8 della JDK e la versione 1.8 della JRE rilasciate da oracle.com. La macchina su cui gira questo Sistema è dotata di 8 GB di RAM e un processore 8core a 4GHZ AMD FX.

2.1 Attori

In questa applicazione sono presenti 2 attori, il **Correntista** e il **Sistema**. L'utente Correntista può eseguire 2 azioni:

- **Prelevare** una certa quantità di denaro dal conto
- **Versare** una certa quantità di denaro nel conto

Il conto sul quale quest'ultimo può operare è istanziato con una somma di denaro già predefinita. Nell'esecuzione del programma possono esistere più correntisti che operano su un unico conto. Si potrebbe ulteriormente migliorare aggiungendo un utente Admin, con maggiori permessi rispetto al correntista, come la possibilità di avere accesso ad ogni movimento di qualunque correntista.

L'utente Sistema è quello che si occupa di stampare la lista movimenti e il saldo quando i correntisti hanno finito il loro operato.

2.2 Casi d'uso

I casi d'uso per questa applicazione sono 2: Versa e Preleva.

Le precondizioni per ambedue i casi d'uso sono le stesse:

- deve essere istanziato un conto con una qualsiasi somma di denaro,
- deve essere istanziato almeno un correntista,
- deve essere avviato almeno un correntista (esecuzione del metodo run() del thread).

Analizziamo nel dettaglio il caso d'uso Versamento:

Al correntista deve essere passata la somma di denaro che dovrà essere versata nel conto, questo prima controlla se il conto non è già occupato da un'altra transazione (se lo è aspetta che viene liberato), poi controlla l'input: se è maggiore di 0 aggiungerà il valore in ingresso alla somma già presente nel conto e aggiungerà questa azione alla lista movimenti, altrimenti comunicherà all'utente un messaggio di errore.

Passiamo al caso d'uso Prelievo che è molto simile al precedente:

Al correntista deve essere passata la somma di denaro che andrà prelevata dal conto, questo controlla che il conto non sia già occupato (in caso lo fosse aspetta che viene liberato), poi controlla l'input: se è maggiore di 0 e minore o uguale alla somma presente nel conto, sottrae l'input al saldo del conto e aggiunge questa azione alla lista movimenti, in caso contrario comunica un messaggio di errore all'utente.

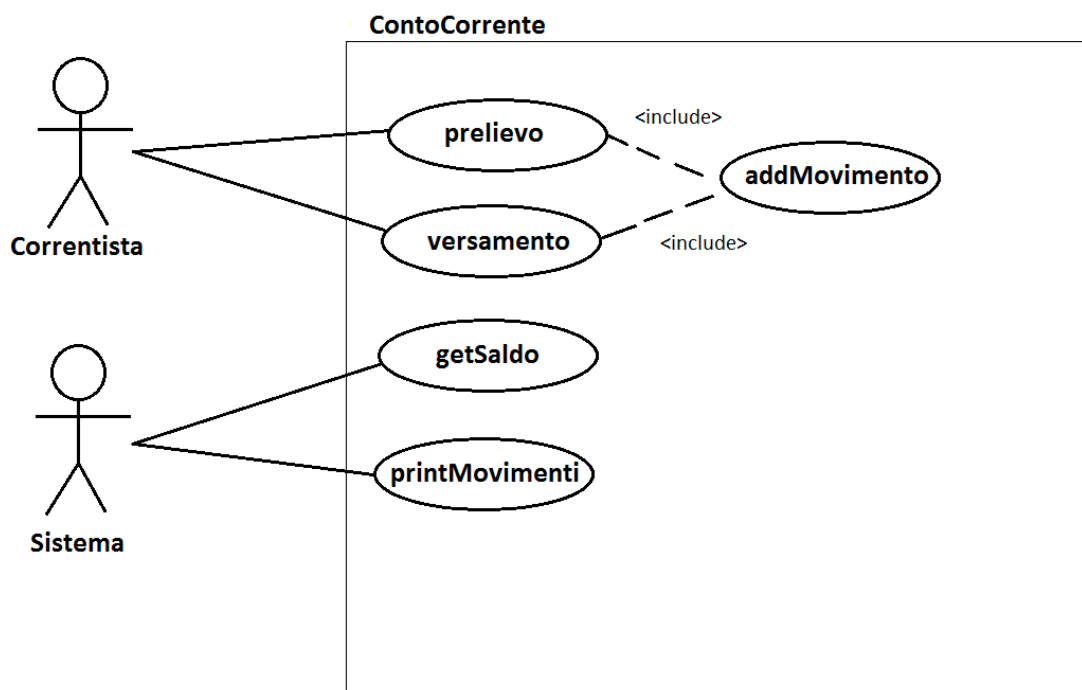
Le postcondizioni per questi casi d'uso sono 2:

- Il conto viene incrementato o decrementato,
- il conto rimane invariato in quanto è sorto un errore nella transazione
-

Eventuali errori che possono insorgere sono:

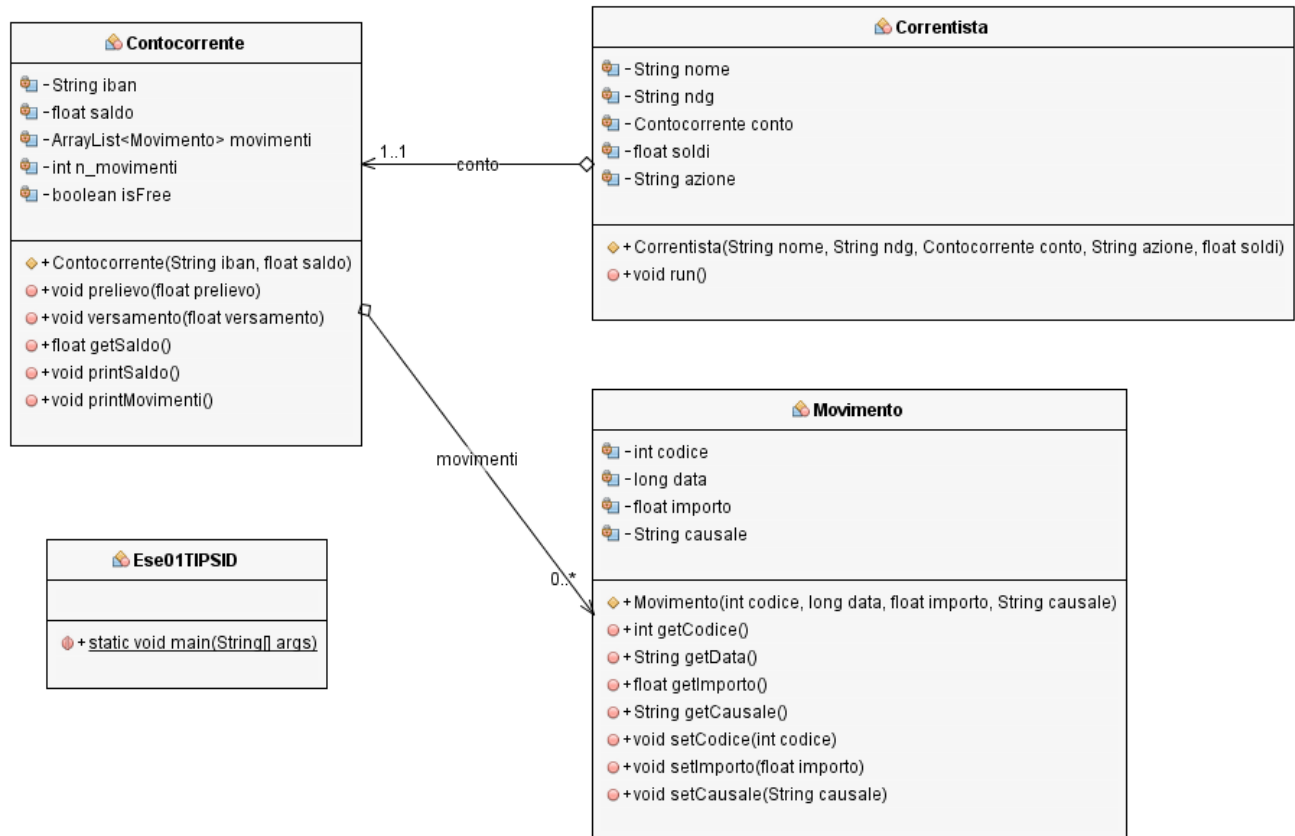
- L'input è minore o uguale a 0,
- la somma da prelevare supera i fondi presenti nel conto.

2.3 Diagramma dei casi d'uso



Documento di progettazione ed implementazione dell'applicazione

1. Diagramma delle classi



5. Commento del codice

Il commento del codice può essere visualizzato tramite questo link: [javadoc](#)

7. Test dell'applicazione

Per testare il codice ho utilizzato una metodologia molto semplice: confrontare l'output dell'esecuzione del codice con il risultato che avevo previsto. Se l'elaborazione del codice corrispondeva al risultato che avevo pensato all'ora tutto era scritto nel modo giusto, altrimenti era sicuramente presente un problema nella sincronizzazione dei threads. Ogni prova è stata ripetuta più volte per assicurarmi che il codice fosse scritto davvero nel modo giusto e che non era solo un caso che l'esecuzione andasse a buon fine.

Ho provato anche a inserire saldi negativi o prelevare una somma maggiore dei soldi presenti nel conto e tutto funziona come dovrebbe.

Licenza e manuale d'uso

1. Licenza d'uso

Open Source

2. Manuale d'uso

[javadoc](#)