

RELAZIONE PROGETTO JAVA DI PR2

FILIPPO RENAI 530478

PARTE 1

1.1

Eccezioni: ho creato tre eccezioni checked per il principio della programmazione difensiva, per distinguere i possibili casi in cui vengono sollevate: *IdErrorException*, *PasswWrongException*, *FileNotFoundException*. Utilizzo la unchecked *NullPointerException* se *Id* o la *passw* o l'oggetto sono uguali a null.

Operazioni aggiuntive:

removeUser: per rimuovere un utente

readF: assumo che *E file* sia una stringa con il path del file che voglio leggere

writeF: assumo che *E file* sia una stringa con il path del file su cui voglio scrivere

(devono essere presente nel Desktop *provaLet* e *provaScritt*, ho usato questi nomi nel test)

Nota: Dal mio punto di vista, il metodo *put*, non ha bisogno d'eccezioni checked essendo di tipo booleano; se l'operazione e' eseguita, venendo rispettati i controlli di identità, con successo ritorna *true*, altrimenti *false* senza sollevare eccezioni.

1.2

Le due implementazioni proposte si basano su due differenti strutture di supporto, la prima sulla classe *ArrayList* (*ArrayListImpl*), mentre la seconda sulla classe *HashMap* (*HashMapImpl*).

La prima implementazione è banale e lenta, la collezione e' 'modellata' come una lista di una classe d'appoggio *UtentiAr* composta dall'*Id*, dalla password, dalla lista dei file posseduti, dalla lista dei file con diritto di lettura e della lista dei file con diritto di lettura e scrittura.

La seconda implementazione, basata su *HashMap*, e' molto piu' ottimizzata e veloce, la collezione e' 'modellata' come una mappa Hash dove le chiavi sono gli *Id* dei singoli utenti e il valore associato e' una classe d'appoggio *UtenteHas* composta come la classe *UtenteAr* senza l'*Id*.

Nei metodi utilizzo direttamente l'oggetto passato nei parametri del metodo per evitare i possibili problemi di aliasing.

PARTE 2

La mia batteria di test e' la classe Test, la main class.

Nella mia batteria utilizzo le due implementazioni, eseguono gli stessi test, per mettere a confronto il loro tempo d' esecuzione; notiamo che l' implementazione con l' HashMap e' piu' efficiente dell'implementazione con ArrayList.

Nel test ho utilizzato tutti i metodi dell'interfaccia, in entrambe l'implementazioni, per ogni metodo faccio prima una chiamata corretta e una chiamata sbagliata per sollevare una delle possibili eccezioni checked.

Non e' mai stata chiamato un metodo con un parametro Null, perche' essendo un'eccezione unchecked non farebbe proseguire il programma, senza un try catch.