



UNIVERSITÀ DI PISA

PROGRAMMAZIONE II

Domenico Profumo | Corso B | Matricola 533695

Il progetto è suddiviso in 5 file:

- **SecureDataContainer**: contiene l'interfaccia della collezione di oggetti di tipo E con le varie funzioni e le specifiche su di esse.
- **MySecureDataContainer**: contiene l'implementazione delle funzioni di SecureDataContainer
- **MySecureDataContainer2**: contiene una seconda implementazione delle funzioni di SecureDataContainer
- **User**: contiene l'implementazione delle funzioni relative alla classe utente
- **Main**: contiene la batteria di test utilizzata per le varie funzioni implementate

MySecureDataContainer

Viene utilizzata un hash map dove associo come chiave il nome utente e come valore associato ad essa l'oggetto utente con i parametri (nome,password,lista dati).

Invocando il metodo costruttore inizializzo la variabile "dim" (contenente il numero di utenti registrati) e creo la funzione per l'hash.

Con **createUser** controllo se i parametri passati come parametro sono !=null e viene fatto un controllo per evitare di aggiungere un utente già registrato. La funzione autenticazione verifica che la password inserita è la stessa di quella dell'utente registrato precedentemente e ritorna l'esito di tale operazione con un valore di verità.

I metodi **put,get,remove,copy** effettuano i controlli di autenticazione dell'utente che le richiede ed effettuano le varie operazioni di inserimento,acquisire,rimuovere e copiare rispettivamente il dato passato come parametro.

La funzione **Getsize** mi da il numero di dati associati all'utente che ne ha fatto richiesta mentre la funzione **share** condivide il dato dell'utente₁ con l'utente₂, la condivisione viene effettuata mediante la copia del dato nella collezione dell'utente₂

(se già non presente in esso), non fa nulla altrimenti. Questa soluzione impedisce di inserire più copie dello stesso dato nella lista dei dati dato che per far ciò vi è una funzione a parte.

Eccezioni: Le eccezioni utilizzate sono principalmente 3, `NullPointerException` quando passo un parametro null alle rispettive funzioni che può essere un nome utente, un dato o una password, `IndexOutOfBoundsException` quando cerco di accedere ad un dato non presente nella collezione dell'utente e `InvalidArgumentException` quando l'utente non è registrato o se la password non è corretta. Sono tutte unchecked, le prime due sono già presenti in java, la terza no.

MySecureData2

Vengono utilizzate due `ArrayList`, la prima memorizza in posizioni diverse i nomi degli utenti registrati, la seconda memorizza al suo interno gli oggetti relativi alla struttura dell'utente associato con i rispettivi parametri di nome, password e dato. Invocando il metodo costruttore inizializzo la variabile "dim" (contenente il numero di utenti registrati) e creo i due `ArrayList`.

I metodi effettuano le stesse operazioni di `MySecureData` differendo solo per alcuni parametri relativi all'implementazione del dato nelle rispettive strutture.

Anche le Eccezioni sono le stesse di quella precedente. In entrambi in casi le eccezioni non vengono gestite all'interno dei metodi ma viene lasciato a chi le utilizza, il compito di gestirle a suo piacere. In questo caso vengono gestite insieme alla batteria dei test nella funzione main.

User

Costituisce la classe utente con i relativi parametri di utente, password e lista dati associati. Invocando il metodo costruttore si inizializzano i parametri relativi all'utente e si va a creare la struttura di supporto per la memorizzazione dei dati mediante un `ArrayList`. Seguono le varie operazioni di inserimento, rimozione, verifica, un iteratore che opera senza modifica dei dati e altre operazioni come dimensione lista dei dati e controllo dell'identità dell'utente.

Le Eccezioni lanciate sono di tipo unchecked già presenti in java ad esempio `IndexOutOfBoundsException`.

Batteria di test

Realizzata nel "Main", andando a testare i metodi di entrambe le implementazioni realizzate, prestando attenzione ai casi limite dove controllo le eccezioni con le clausole try e catch in modo da poterle gestire.

Inizialmente registro i vari utenti, controllo la corretta registrazione, inserisco i dati in ciascuno di essi, effettuo operazioni di copia dei dati e di condivisioni tra utenti andando poi a stampare a schermo attraverso un iteratore la lista dei dati presenti nell'utente che ha fatto richiesta fino a rimuoverli e a cancellarli.