

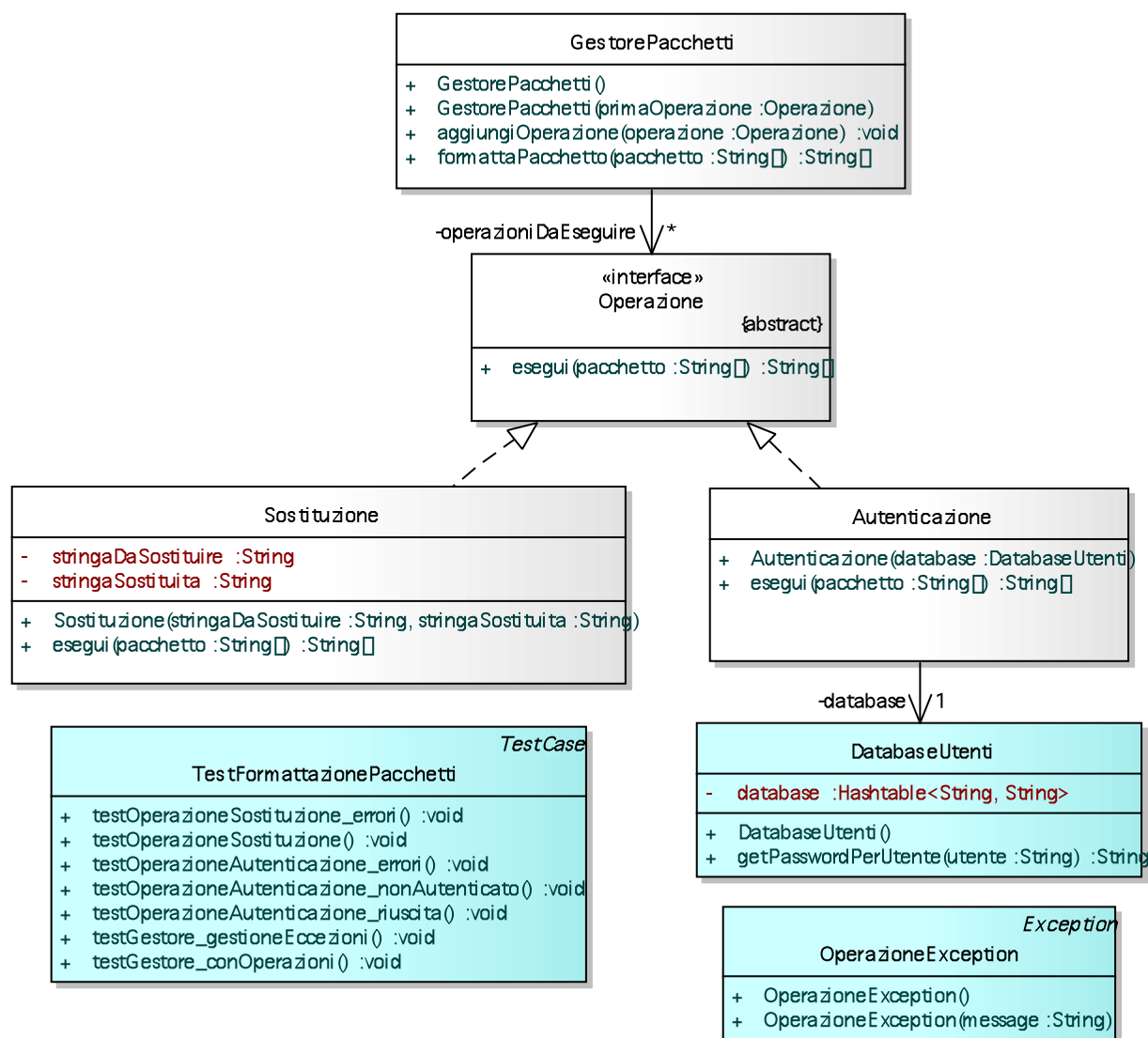
Programmazione 2

Testo parte di pratica

Si realizzi un “sistema di formattazione di pacchetti di testo” come da diagramma UML e specifiche che seguono. Si provino le classi realizzate con JUnit utilizzando i test forniti nella classe TestFormattazionePacchetti.

Nota: Gli elaborati che non superino almeno 2 test fra quelli dati saranno considerati insufficienti.

(Suggerimento: si eviti di eseguire i test solo alla fine del lavoro, quando ormai sarebbe tardi per apportare correzioni; Piuttosto si eseguano i test man mano che si procede con l’implementazione, per verificare incrementalmente il lavoro via via fatto.)



Interfaccia Operazione:

- Rappresenta l’interfaccia di tutte le operazioni che possono essere eseguite su un pacchetto di testo. Nel contesto di quest’esercizio, per “pacchetto di testo” si intende sempre un “array di stringhe”.
- Definisce il metodo `esegui(String[] pacchetto) : String[]` che riceve in input un pacchetto, lo manipola secondo un’operazione (che sarà implementata dalle classi che implementano l’interfaccia), e restituisce come valore di ritorno il pacchetto risultante dopo l’operazione.

- Si tenga presente il modo in cui l'interfaccia `Operazione` viene usata polimorficamente nei test forniti.

Classe Sostituzione:

- Realizza un'operazione che identifica, fra le stringhe presenti nel pacchetto di input, quelle uguali a una stringa `data` (`StringaDaSostituire`) e le sostituisce con una stringa diversa (`StringaSostituita`).
- Il costruttore inizializza i valori `StringaDaSostituire` e `StringaSostituita` in base ai parametri.
- Il metodo `esegui(String[] pacchetto): String[]` applica l'operazione di sostituzione alle stringhe nel pacchetto in input e restituisce il pacchetto così modificato. Nel caso in cui il parametro sia `null`, il metodo fallisce sollevando un'eccezione di tipo `OperazioneException`.

(La classe `OperazioneException` viene fornita già implementata e non deve essere modificata.)

Classe Autenticazione:

- Realizza un'operazione di autenticazione in base alle credenziali utente contenute in un pacchetto.
- Il costruttore inizializza l'associazione fra l'oggetto `Autenticazione` e un oggetto `DatabaseUtenti` passato come parametro. (La classe `DatabaseUtente` viene fornita già implementata e non deve essere modificata).
- Il metodo `esegui(String[] pacchetto): String[]` effettua l'operazione di autenticazione, ovvero:
 - Verifica che il pacchetto in input contenga almeno due stringhe. In caso contrario, l'operazione fallisce sollevando un'eccezione di tipo `OperazioneException`. Suggerimento: l'eccezione deve essere sollevata se si verifica uno dei seguenti casi: 1) il parametro è `null`; 2) l'array in input ha una lunghezza minore di due; 3) uno delle prime due stringhe dell'array è `null`.
 - Estrae le prime due stringhe dell'array. Secondo quest'operazione, la prima estratta dall'array rappresenta il nome di un utente e la seconda rappresenta la sua password.
 - Confronta la password dell'utente con quella ottenuta invocando il metodo `DatabaseUtenti.getPasswordUtente`.
 - Se la password non è uguale a quella nel database, solleva un'eccezione di tipo `OperazioneException`. Se invece è uguale, restituisce il pacchetto che si ottiene da quello in input dopo aver rimosso le prime due stringhe.

Classe GestorePacchetti:

- Definisce un modulo in grado di formattare pacchetti applicando a essi una sequenza di operazioni. L'associazione fra l'oggetto `GestorePacchetti` gli oggetti `Operazione` è realizzata con una collezione di tipo `ArrayList`.
- Il costruttore `GestorePacchetti()` inizializza l'oggetto associandolo con una collezione vuota.
- Il costruttore `GestorePacchetti(Operazione)` inizializza l'oggetto associandolo con la singola operazione passata come parametro.
- Il metodo `aggiungiOperazione(Operazione)` aggiunge una nuova operazione alla collezione.
- Il metodo `formattaPacchetto(String[] pacchetto): String[]` modifica il pacchetto in input applicando, in sequenza, tutte le operazioni della collezione. Se una qualsiasi delle operazioni fallisce, il metodo restituisce `null`. Altrimenti, il metodo restituisce il pacchetto ottenuto dopo aver applicato tutte le operazioni.