StringPool:
La classe StringPool

Questa dispensa fornisce le informazioni necessarie per l'implementazione della prima versione della classe StringPool. Lo scopo dell'esercitazione è di acquisire la giusta pratica nella scrittura corretta di codice e nella sua documentazione.

Specifica

La classe StringPool deve consentire la gestione di un insieme di oggetti di tipo `String` (d'ora in avanti: stringhe) collegate ad identificatori rappresentati mediante numeri interi. Ad esempio, un oggetto di tipo StringPool potrà contenere elementi come segue:

<125, “Questa è una stringa”>

<12349, “Questa è un'altra stringa”>

…

Chiameremo “identificatore” il numero associato ad una stringa. Le operazioni di gestione di questi elementi devono garantire che in un oggetto StringPool non vi siano mai due elementi con lo stesso identificatore.

Servizi offerti

La classe StringPool deve fornire i servizi seguenti, attraverso metodi di classe.

Inserimento di un elemento

Questo servizio consente di aggiungere in un oggetto StringPool una nuova stringa associata ad un numero intero. L'inserimento ha successo solo se l'identificatore specificato non è già presente nel contenitore.

Input

Identificatore: intero positivo

Stringa: oggetto di tipo String, non null.

Output

Un valore Booleano che indica:

- TRUE se l'inserimento ha avuto successo;
- FALSE se l'inserimento non ha avuto successo.

Rimozione di un elemento

Questo servizio consente di rimuovere un elemento dalla collezione, quando se ne specifica l'identificatore. La rimozione ha successo solo se l'identificatore specificato corrisponde a qualche elemento nel contenitore.

Input

Identificatore: intero positivo

Output

Un valore Booleano che indica:

- TRUE se la rimozione ha avuto successo
- FALSE se la rimozione non ha avuto successo

Sostituzione di un elemento

Questo servizio corrisponde alla rimozione di un elemento con identificatore specificato, e al successivo inserimento dell'elemento specificato. Valgono le condizioni necessarie per la rimozione di un elemento e per il successivo inserimento di un elemento.

Input

Identificatore: intero positivo

Stringa: `String`

Output

Un valore Booleano che indica:

- TRUE se la sostituzione ha successo;
- FALSE se la sostituzione non ha avuto successo.

Rappresentazione in stringa

Questo servizio consente di ottenere una rappresentazione a stringa dell'intero contenitore. Per questioni tecniche legate alla gerarchia delle classi Java, il nome di questo servizio sarà necessariamente “toString” e non richiederà input.

Il formato di rappresentazione richiesto è indicato dal seguente esempio:

```
{
      <125, Questa è una stringa>
      <12349, Questa è un'altra stringa>
}
```

Specifica della classe

Sulla base della specifica dei requisiti indicati sopra, si specifica nel dettaglio l'interfaccia esposta dalla classe StringPool.

Metodi di classe

`add (int identifier, String str) → Boolean`

`remove (int identifier) → Boolean`

`replace (int identifier, String str) → Boolean`

`toString () → String`

Progettazione della classe

Membri privati

La classe StringPool include due array per il contenimento degli elementi. Il primo array conterrà le stringhe, mentre il secondo gli identificatori. Un elemento è dato dalla coppia (identificatore, stringa) nella medesima posizione.

Campo `strings`

Il campo `strings` è un array di stringhe. La dimensione dell'array è specificata come parametro del costruttore. In fase di inizializzazione, tutti gli elementi conterranno l'oggetto null.

Campo `identifiers`

Il campo `identifiers` è un array di interi. La dimensione dell'array è specificata come parametro del costruttore. In fase di inizializzazione, tutti gli elementi conterranno il valore -1, che rappresenta lo stato di assenza dell'elemento nella posizione corrispondente.

Metodo `elementToString`

Questo metodo privato consente di ottenere una rappresentazione in stringa di un singolo elemento in posizione specificata

Input

pos : `int`

Output

Una stringa nel formato:

<*identificatore*, *stringa*>

dove `identificatore` corrisponde alla rappresentazione in stringa dell'identificatore memorizzato nella posizione `pos` dell'array `identifiers`, e `stringa` corrisponde alla stringa memorizzata in `strings` nella posizione `pos`.

Membri pubblici

I membri pubblici corrispondono ai servizi specificati nella sezione precedente, compreso il costruttore della classe.

Costruttore

Il costruttore accetta un parametro intero che indica la dimensione degli array `strings` e `identifiers`.

Il corpo del costruttore crea i due array, assicurandosi che tutti gli elementi di `strings` sono inizializzati a `null` e tutti gli elementi di `identifiers` siano inizializzati a -1.

Metodo add

Il metodo `add` riceve in input un identificatore ed una stringa.

Il metodo dapprima verifica se gli input soddisfano le specifiche:

- `identifier > 0`
- `str != null`

Se gli input non soddisfano i requisiti, il metodo termina restituendo FALSE.

Successivamente, effettua una ricerca per verificare la presenza dello stesso identificatore nell'array `identifiers`. Se la ricerca ha successo, la procedura termina restituendo FALSE.

Infine, il metodo ricerca la prima posizione di `identifiers` con valore -1. Se non vi sono elementi con questo valore, il metodo termina restituendo FALSE. Altrimenti memorizza nella posizione trovata l'identificare di input, e nella posizione corrispondente di `strings` la stringa di input.

Metodo `remove`

Il metodo `remove` riceve in input un identificatore.

Il metodo effettua una ricerca sull'array `identifiers` e, se trova un valore corrispondente all'identificatore di input, modifica questo valore a -1, pone a `null` il corrispondente valore dell'array `strings` e restituisce TRUE. Se la ricerca non ha successo restituisce FALSE.

Metodo `replace`

Il metodo `replace` riceve in input un identificatore ed una stringa.

Il metodo dapprima richiama il metodo `remove` per eliminare la stringa. Se l'eliminazione ha successo chiama il metodo `add` per aggiungere l'elemento specificato.

Il metodo restituisce TRUE se e solo se entrambi i metodi `remove` e `add` restituiscono TRUE, altrimenti restituisce FALSE.

Metodo `toString`

Il metodo `toString` restituisce un oggetto di tipo `String`. Esso prepara una stringa inserendo dapprima il carattere '{', poi una newline. Esso cicla su `identifiers` e per ogni valore diverso da -1 richiama il metodo privato `elementToString` per ottenerne la rappresentazione in stringa da giustapporre alla stringa finale. Dopo ogni inserimento inserisce una newline. Al termine del ciclo giustappone il carattere '}'. La stringa finale viene restituita in output.

Implementazione

L'implementazione è lasciata allo studente.

Occorre creare la classe `StringPool` *seguendo le specifiche sopra riportate e attenendosi alle convenzioni di scrittura e di documentazione viste a lezione.*

Aggiungere alla classe il metodo `main` *per testare la correttezza del programma.*

Inizialmente il metodo crea un oggetto di tipo `StringPool` *con 20 elementi e di nome* `sp`.

Il programma acquisisce ciclicamente una stringa da tastiera e genera casualmente un numero compreso tra 1 e 10 (usare la classe java.util.Random). Ad ogni ciclo la procedura tenta di inserire la stringa in `sp`. *Se l'inserimento non ha successo il sistema chiede all'utente se vuole terminare o provare ad effettuare una sostituzione anziché un inserimento. Il sistema procederà in accordo con la richiesta dell'utente. Il ciclo termina quando l'utente decide di terminare l'inserimento.*