

TASK

NokiaPhonebookTask

Write a class in Java or C++ language that meets the following requirements.

It is a phonebook. It should contain phone numbers and subscribers. For any subscriber, one or more phone numbers can be assigned. For any phone number exactly one subscriber can be assigned.

Provide search by phone number function (returns a string or throws an exception).

Provide search by subscriber function (returns a list of phone numbers).

Make both searches as efficient as possible.

The internal representation is up to you, but please do not use any data structures provided by the language or frameworks. Arrays can be used, enums, classes, records, pointers can be used.

Provide a function that expects two parameters (name and phone number) and adds these to your phonebook as a pair. It should throw an exception when the exact same record is already in the phonebook, or if the input is malformed.

Provide another function with these parameters. These function will delete the given record from the phonebook. If no such entry, raise an exception.

Subscriber's name could be anything. Accepted phone number format (regex) should be configured externally (e.g. as a command line argument).

Create a command line app or a unit test set to present the features of your class.

Write clean, well-organized, self documenting code.

Phonebook Documentation

The PhonebookDataType class

The PhonebookDataType class is a data structure used to store phonebook entries. It has two private fields, subscriber and phoneNumbers, which store the subscriber's name and their phone numbers, respectively. The class provides several public methods to interact with the stored data:

- GetSubscriber() returns the subscriber's name.
- GetPhoneNumbers() returns an array of strings that represent the subscriber's phone numbers.
- AddPhoneNumber(String phoneNumber) increases the size of the phoneNumbers array and adds the new phone number to the end of the array.
- DeletePhoneNumber(String phoneNumber) deletes the given phone number from the phoneNumbers array.

The class has a constructor that takes a subscriber's name and a phone number as parameters and creates a new PhonebookDataType object with the given subscriber and phone number. The class uses an array of strings to store the phone numbers as one subscriber can have multiple phone numbers. The AddPhoneNumber method uses the ArrayUtils.increaseArraySize method to dynamically increase the phoneNumbers array size when adding a new phone number. The DeletePhoneNumber method implements the logic for deleting a phone number from the phoneNumbers array.

Phonebook Class Documentation

The Phonebook class is a Java class that provides basic functionalities for a phonebook application. It allows adding, searching, and deleting entries from a phonebook. Each phonebook entry consists of a subscriber name and one or more phone numbers.

Class Members

- PhonebookDataType[] phonebookEntries

An array of PhonebookDataType objects representing phonebook entries.

- Pattern phoneNumberPattern

A pattern object to validate the format of phone numbers.

Constructor

- Phonebook(String phoneNumberRegex)

This constructor initializes an instance of the Phonebook class with an empty phonebook entries array and sets a pattern for phone number validation. The phoneNumberRegex parameter specifies the pattern to use for phone number validation.

Methods

- `boolean isPhoneNumberValid(String phoneNumber)`

This method validates the format of a phone number by checking if it matches the phone number pattern set in the constructor.

- `boolean isPhoneNumberExists(String phoneNumber)`

This method checks if the given phone number exists in the phonebook entries.

- `boolean isSubscriberExists(String subscriber)`

This method checks if the given subscriber name exists in the phonebook entries.

- `void addEntry(String subscriber, String phoneNumber) throws Exception`

This method adds a new phonebook entry with the given subscriber name and phone number. It validates the phone number format and throws an exception if it's invalid. If the phone number already exists in the phonebook, it throws an exception. If the subscriber already exists in the phonebook, it adds the phone number to the existing subscriber's entry.

- `String searchPhonebookByPhoneNumber(String phoneNumber)`

This method searches the phonebook entries for a subscriber with the given phone number and returns the subscriber's name. If the phone number is not found in the phonebook, it returns null.

- `void deleteEntry(String subscriber, String phoneNumber) throws Exception`

This method deletes a phonebook entry with the given subscriber name and phone number. If the phone number does not exist in the phonebook, it throws an exception. If the subscriber does not exist in the phonebook, it throws an exception.

- `PhonebookDataType[] deleteArrayElement(PhonebookDataType[] entries, int index)`

This method deletes an element from an array of PhonebookDataType objects. The index parameter specifies the index of the element to be deleted. If the index is out of range, it throws an `ArrayIndexOutOfBoundsException`.

ArrayUtils Class

The ArrayUtils class is a utility class that provides methods for manipulating arrays in Java. It provides two main functions:

- `increaseArraySize(T[] arr, int newSize)`

This method increases the size of an array by creating a new array with the new size and copying the elements from the old array into the new one.

The method takes in two parameters:

`arr`: The array to be increased in size.

`newSize`: The new desired size of the array.

The method returns a new array with the increased size.

- `deleteArrayElement(T[] array, int indexToRemove)`

This method deletes an element from an array by creating a new array without that element and copying the remaining elements from the old array into the new one.

The method takes in two parameters:

`array`: The array from which the element will be deleted.

`indexToRemove`: The index of the element to be removed.

The method returns a new array with the removed element.

If the index provided is out of bounds, the method will throw an `ArrayIndexOutOfBoundsException`.

PhonebookApp Class

The PhonebookApp class is an application that implements a phonebook to store and manage phone numbers for different subscribers. It uses the Phonebook class to perform all its operations, including adding and deleting entries, searching for a subscriber by their name, and searching for a subscriber by their phone number.

The PhonebookApp class uses a Scanner object to take inputs from the user, and a while loop to present a menu to the user that allows them to perform different actions with the phonebook. The menu has five options:

- **Add Phonebook Entry** - The user can add a new entry to the phonebook by entering the subscriber's name and their phone number. If the phone number does not match the phone number format defined in the Phonebook class, an exception will be thrown.
- **Delete Phonebook Entry** - The user can delete an existing entry from the phonebook by entering the subscriber's name and their phone number.

- Search Phonebook by Subscriber - The user can search for a subscriber by entering their name. The phonebook will return all phone numbers associated with the subscriber.
- Search Phonebook by Phone Number - The user can search for a subscriber by entering their phone number. The phonebook will return the subscriber's name if the phone number is found. If the phone number is not found, the user will be notified.
- Exit - The user can exit the application by selecting this option.

The PhonebookApp class uses a switch statement to handle the user's choice and perform the appropriate action. The switch statement has a default case that is executed when the user enters an invalid choice.

Finally, when the user exits the application, the Scanner object is closed and the program terminates.