

Bereinigung von Telefonnummern

Einführung

Du arbeitest in einer Firma mit einer riesigen Anzahl von Telefonnummern. Leider sind die Daten ein großes Durcheinander. Sie wurden in verschiedenen Formen eingegeben (z.B. +43722912345 oder 0660-75099345 oder 0043/1/034508). Es ist deine Aufgabe, ein Befehlszeilen-Hilfsprogramm zu implementieren, das die Telefonnummern analysiert.

Grundlegende Anforderungen

- Implementiere eine Befehlszeilenanwendung in C#.
- Die Anwendung erhält eine Telefonnummer als Befehlszeilenargument. Wenn das Programm ohne Argument aufgerufen wird, muss es eine Fehlermeldung ausgeben (*Missing phone number, please try again*).
- Etwaige Schrägstriche (/) oder Bindestriche (-) in Telefonnummern müssen ignoriert werden.
- Dein Programm muss Telefonnummern basierend auf folgendem Regelwerk prüfen:
 - Eine Telefonnummer könnte eine Notrufnummer sein. Gültige Notrufnummern sind 112, 122, 133 und 144.
 - Wenn die Telefonnummer keine Notrufnummer ist, ...
 - * ...muss sie mit +43 oder 0043 oder 0 beginnen und
 - * ...müssen alle verbleibenden Telefonnummernzeichen Ziffern sein und
 - * ...muss die Länge der Telefonnummer mindestens 6 Ziffern betragen.
- Wenn die Telefonnummer eine Notrufnummer ist, muss dein Programm *Emergency number: {Telefonnummer}* ausgeben.
- Wenn die Telefonnummer ungültig ist, muss dein Programm *Invalid phone number: {Telefonnummer}* ausgeben.

- Wenn die Telefonnummer gültig und keine Notrufnummer ist, muss dein Programm die Telefonnummer mit führendem +43 ausgeben. Hier sind einige Testdaten, die zeigen, wie dein Programm Telefonnummern umwandeln muss:

Eingabe	Ausgegebene Telefonnummer
+4373212345	+4373212345
004373212345	+4373212345
073212345	+4373212345

- Du musst eine Methode `IsValidPhoneNumber` implementieren und verwenden. Diese Methode erhält die Telefonnummer aus dem Befehlszeilenargument als Eingabeparameter und gibt `true` zurück, wenn die Nummer nach den oben definierten Regeln gültig ist, sonst `false`.
- Du musst eine Methode `CleanupPhoneNumber` implementieren und verwenden. Diese Methode erhält die Telefonnummer aus dem Befehlszeilenargument und gibt die bereinigte Version wie in der Tabelle oben gezeigt zurück. Wenn du möchtest, kannst du einen `ref`-Parameter anstelle der Rückgabe der bereinigten Telefonnummer verwenden.

Erweiterte Anforderungen

- Deine Anwendung muss in der Lage sein, **eine oder mehrere Telefonnummern** in der Befehlszeile zu verarbeiten. Wenn das Programm ohne Argument aufgerufen wird, muss es eine Fehlermeldung ausgeben (*Missing phone number(s), please try again*).
- Zusätzlich zur bereinigten Telefonnummer muss dein Programm getrennt die Vorwahl und die Telefonnummer ohne Vorwahl ausgeben. Hier sind die Regeln zur Extraktion von Vorwahlen:
 - Wien hat die einstellige Vorwahl 1.
 - Die folgenden Vorwahlen haben drei Ziffern: 316, 512, 463, 732, 662, 660, 699, 664, 676, 650, 680, 681, 665, 688, 677, 670, 690
 - Alle anderen Vorwahlen haben vier Ziffern.
 - Notrufnummern haben keine Vorwahl.

- Du musst eine Methode `SplitPhoneNumber` implementieren und verwenden. Diese Methode erhält die Telefonnummer aus dem Befehlszeilenargument und gibt Vorwahl und Telefonnummer ohne Vorwahl zurück. Verwende `out`-Parameter, um beide Werte zurückzugeben.

Hier sind einige Testdaten mit Eingabe und entsprechender Ausgabe:

Eingabe	Erwartete Ausgabe
112	Emergency number: 112
+4312abc	Invalid phone number: +4312abc
0043-1-9876543	+4319876543, Vorwahl 1, Telefonnummer 9876543
+43/732/12345	+4373212345, Vorwahl 732, Telefonnummer 12345
07229234567	+437229234567, Vorwahl 7229, Telefonnummer 234567
0123	Invalid phone number: 0123
01234567	+431234567, Vorwahl 1, Telefonnummer 234567