E-Mail-Parser

Einleitung

Du arbeitest in einem Unternehmen mit einer großen CRM-Datenbank (Customer Relationship Management). Sie enthält viele E-Mail-Adressen. Leider sind viele davon falsch. Die Daten sind ein großes Durcheinander. Es ist deine Aufgabe, ein Befehlszeilen-Hilfsprogramm zu implementieren, das E-Mail-Adressen analysiert und überprüft.

Grundlegende Anforderungen

- Implementiere eine Befehlszeilenanwendung in C#.

- Die Anwendung erhält eine E-Mail-Adresse als Befehlszeilenargument. Wenn das Programm ohne Argument aufgerufen wird, muss es eine Fehlermeldung ausgeben (*Fehlende E-Mail-Adresse, bitte versuche es erneut*).

- Alle Großbuchstaben (*A-Z*) müssen vor der weiteren Verarbeitung in Kleinbuchstaben (*a-z*) umgewandelt werden.

- Eine gültige E-Mail-Adresse (z.B. *r.stropek@htl-leonding.ac.at*) besteht aus den folgenden Teilen:

  1. E-Mail-Präfix (z.B. *r.stropek*)

  2. Ein *@* Zeichen

  3. Domain (z.B. *htl-leonding.ac.at*)

- Dein Programm muss E-Mail-Adressen anhand der folgenden Regeln prüfen:

  - E-Mail-Präfixe können Kleinbuchstaben (*a-z*), Ziffern (*0-9*), Unterstriche (*_), Punkte (.) und Bindestriche (-*) enthalten.

  - Domains können Kleinbuchstaben (*a-z*), Ziffern (*0-9*), Punkte (.) und Bindestriche (-) enthalten.

- Wenn die E-Mail-Adresse ungültig ist, muss dein Programm *Ungültige E-Mail-Adresse: {E-Mail-Adresse}* ausgeben.

- Wenn die E-Mail-Adresse gültig ist, muss dein Programm das E-Mail-Präfix und die Domain ausgeben. Hier sind einige Testdaten, die Eingabe und Ausgabe veranschaulichen:

Eingabe	Ausgabe
r.stropek@htl-leonding.ac.at	E-Mail r.stropek@htl-leonding.ac.at besteht aus Präfix r.stropek und Domain htl-leonding.ac.at
fred@flintstones.com	E-Mail fred@flintstones.com besteht aus Präfix fred und Domain flintstones.com
magic_chris.w1z@wizzar 123.com	E-Mail magic_chris.w1z@wizzard-123.com besteht aus Präfix magic_chris.w1z und Domain wizzard-123.com
abc	Ungültige E-Mail-Adresse: abc
abc@@def.com	Ungültige E-Mail-Adresse: abc@@def.com
abc@def@ghi.com	Ungültige E-Mail-Adresse: abc@def@ghi.com
abc@	Ungültige E-Mail-Adresse: abc@
@def.com	Ungültige E-Mail-Adresse: @def.com
abc..d@efg.com	E-Mail abc..d@efg.com besteht aus Präfix abc..d und Domain efg.com
abc@def.comm	E-Mail abc@def.comm besteht aus Präfix abc und Domain def.comm

- Du musst eine Methode `IsValidEmailAddress` implementieren und verwenden. Diese Methode erhält die E-Mail-Adresse aus dem Befehlszeilenargument als Eingabeparameter und gibt true zurück, wenn die Adresse nach den oben definierten Regeln gültig ist, andernfalls false.

- Du musst eine Methode `AnalyzeEmailAddress` implementieren und verwenden. Diese Methode erhält die E-Mail-Adresse aus dem Befehlszeilenargument und gibt die Ausgabe (siehe Tabelle oben) als Zeichenkette zurück.

- Du kannst zusätzliche Methoden hinzufügen, wenn du möchtest.

Erweiterte Anforderungen

- Deine Anwendung muss in der Lage sein, **eine oder mehrere E-Mail-Adressen** in der Befehlszeile zu verarbeiten. Wenn das Programm ohne Argument aufgerufen wird, muss es eine Fehlermeldung ausgeben (*Fehlende E-Mail-Adresse(n), bitte versuche es erneut*).

- Es gibt zusätzliche Validierungsregeln:
  - Präfix: Ein Unterstrich, Punkt oder Bindestrich muss von mindestens einem Buchstaben oder einer Zahl gefolgt werden.

  - Domain: Der letzte Teil der Domain muss mindestens zwei Zeichen lang sein, zum Beispiel: *.com*, *.org*, *.cc*

Die oben gezeigten Eingabe-/Ausgabe-Kombinationen müssen zum gleichen Ergebnis führen wie zuvor, **mit Ausnahme** der beiden letzten Beispiele. Sie sind nun ungültig:

Eingabe	Ausgabe
abc..d@efg.com	Ungültige E-Mail-Adresse: abc..d@efg.com
abc@def.comm	Ungültige E-Mail-Adresse: abc@def.comm
abc@def.c	Ungültige E-Mail-Adresse: abc@def.c