Dokumentation:

Algorithmus: Rekursive Bestimmung von Primzahlen.

* Ausgelagert in MyPrimeHelper.cs
* istPrimzahl(int) prüft ob übergabe Zahl kleiner gleich 1 ist
  + ja: keine Primzahl
  + nein: starte rekursive Primzahlbestimmung
* istPrimzahl(int, int) prüft ob gesuchte zahl durch temporäre zahl teibar ist
  + ja: wenn gesuchte zahl == temporäre zahl keine primzahl sonst primzahl
  + nein: aufruf der eigenen funktion mit temporäre zahl +1

Containerverwendung:

* Interface MyIdentify -> Id
* Container können nur Objekte aufnehmen, die das Interface MyIdentify implementieren.
* MyList.cs
  + Erbt von Liste
  + Implementiert MyIdentify
  + Besitzt Methode getPrimes zur primzahlbestimmung
* MyQueue.cs
  + Erbt von Queue
  + Implementiert MyIdentify
  + Besitzt Methode getPrimes zur primzahlbestimmung

Erzeugung Eingangsdaten

* Daten werden in einer csv Datei gespeichert
* Trennung durch Semikolon

Limitationen der Eingangsdaten

* Pokemon
  + ID integer
  + Name string
  + Type String
  + Legendary boolean
* Digimon
  + Id integer
  + Name String
  + Health String
* Objekte dürfen keine ID kleiner 0 sein sonst individueller Fehler
* Eingabe Datei muss diese Reihenfolge beinhalten