CSV PROCESSING IMPROVEMENTS

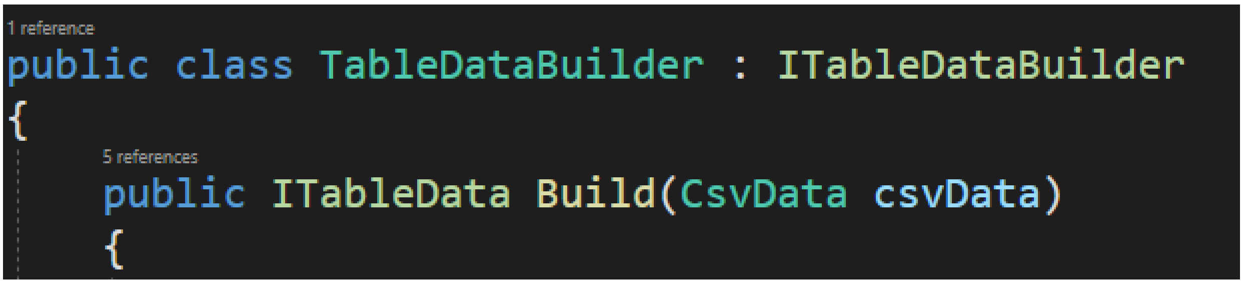
Ultimate C# Masterclass Assignment

Overview

The purpose of this assignment is to improve the performance of a library processing CSV files. The aim is to make this code work faster and consume less memory.	Library

Existing code

The project to be refactored can be found in the Git repository, as well as in the resources of the “Assignment - CSV Processing Improvements - Description and requirements” lecture. The project already has the sampleData.csv file present in its bin folder.



```
public class TableDataBuilder : ITableDataBuilder
{
    public ITableData Build(CsvData csvData)
    {
```

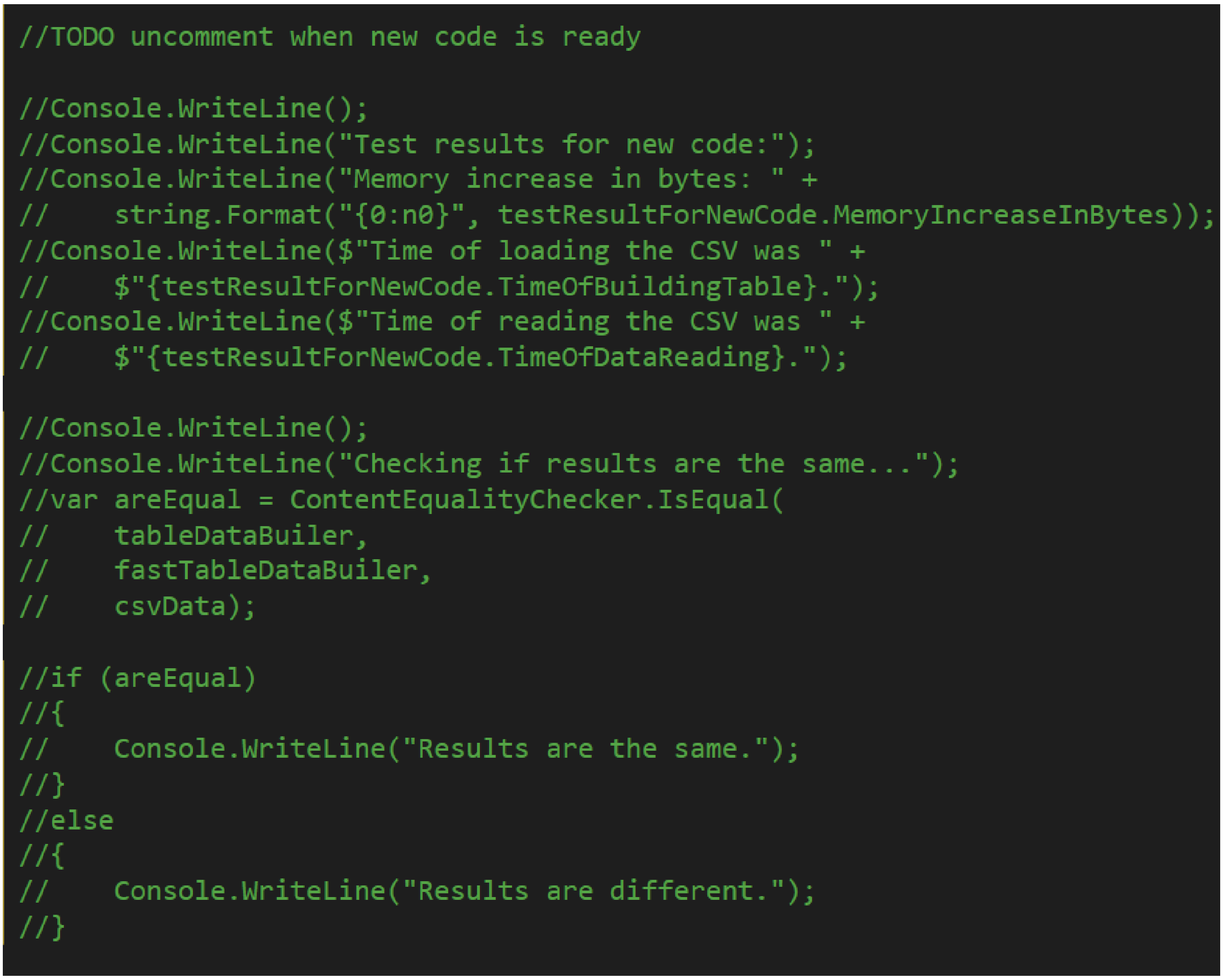
The core of this project is the TableDataBuilder class, which builds a TableData object. This object exposes properties for reading column names and the count of rows in the table, as well as a method for reading the value of a specific cell (returned as an object since this table can store various types of data).

Purpose

Create an alternative implementation of the TableDataBuilder class and the ITableData interface. Try to make the new version consume less memory and work faster.

Performance measurements

The project already contains code that can measure both execution time and memory consumption. To test your version of the code, uncomment the following lines in the Program class:



```
//TODO uncomment when new code is ready

//Console.WriteLine();
//Console.WriteLine("Test results for new code:");
//Console.WriteLine("Memory increase in bytes: " +
//    string.Format("{0:n0}", testResultForNewCode.MemoryIncreaseInBytes));
//Console.WriteLine($"Time of loading the CSV was " +
//    $"{testResultForNewCode.TimeOfBuildingTable}.");
//Console.WriteLine($"Time of reading the CSV was " +
//    $"{testResultForNewCode.TimeOfDataReading}.");

//Console.WriteLine();
//Console.WriteLine("Checking if results are the same...");
//var areEqual = ContentEqualityChecker.IsEqual(
//    tableDataBuiler,
//    fastTableDataBuiler,
//    csvData);

//if (areEqual)
//{
//    Console.WriteLine("Results are the same.");
//}
//else
//{
//    Console.WriteLine("Results are different.");
//}
```

Assumptions

- Don't modify any existing code.
- The input CSVs are usually big (the sample file has 430 columns and 4000 rows, and we should expect larger files to be processed too).
- The input CSVs only contain bool, int, decimal, and string columns.
- The input CSVs can have empty cells (and usually have many).