

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота № 5**

з дисципліни “Основи програмування”

тема “Рядки символів та файли”

|  |  |  |
| --- | --- | --- |
| Виконал  студент I курсу  групи КП-02  Жученко Андрій Сергійович  варіант №5 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2020

**Мета роботи**

Навчитися працювати з рядками символів.

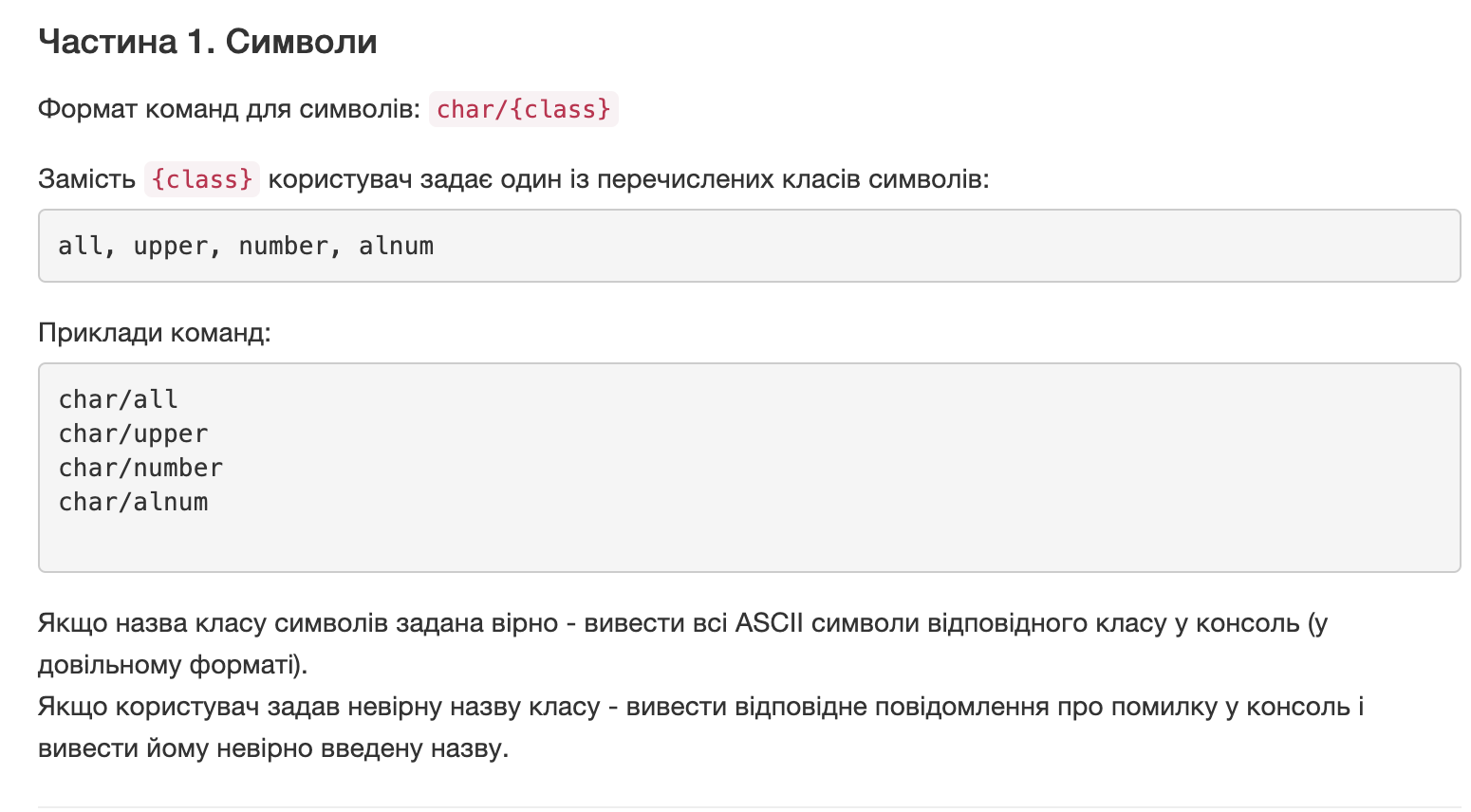
Вміти визначати класи символів та використовувати функції для роботи з символами.

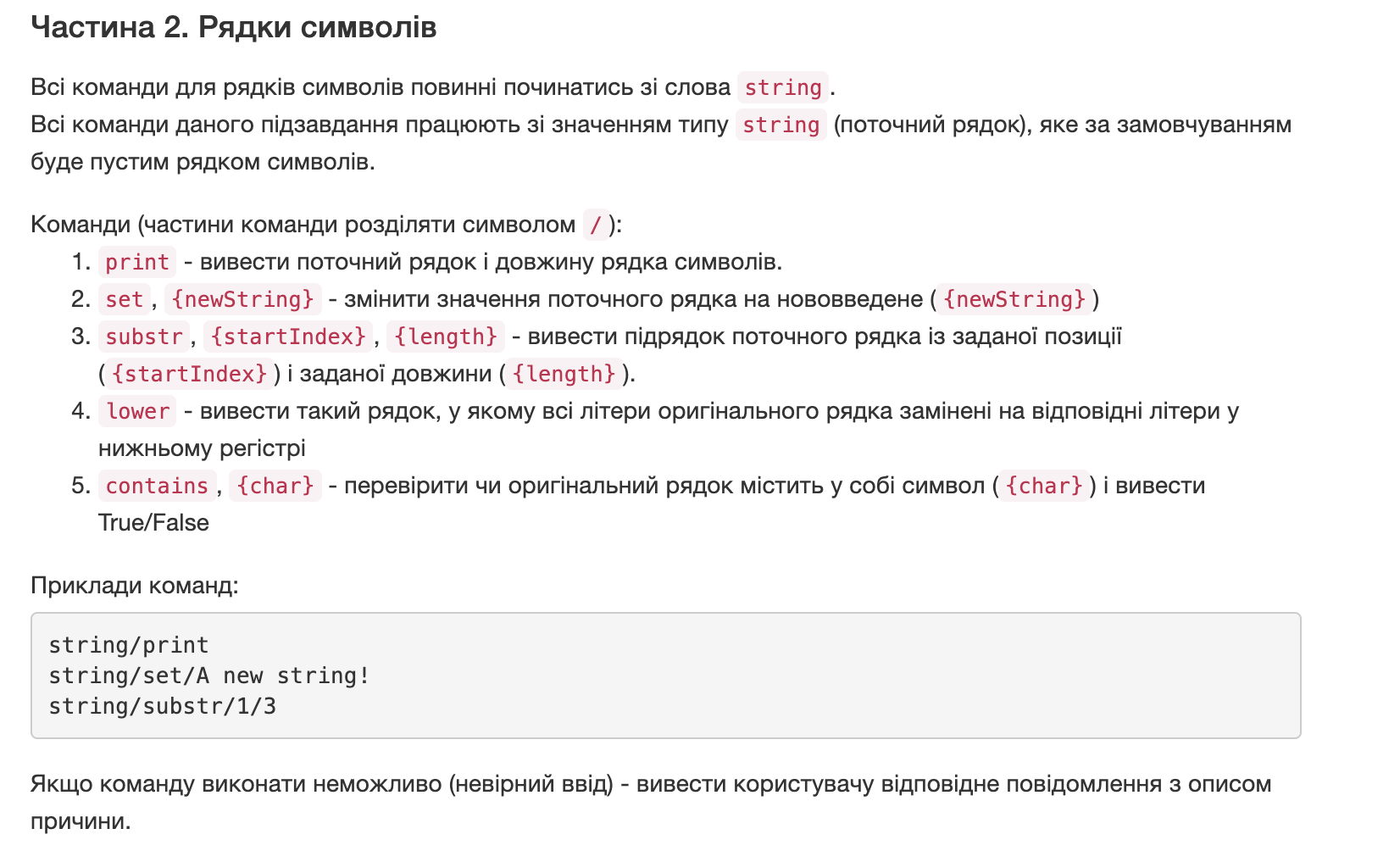
Навчитись використовувати стандартні функції для роботи з рядками символів.

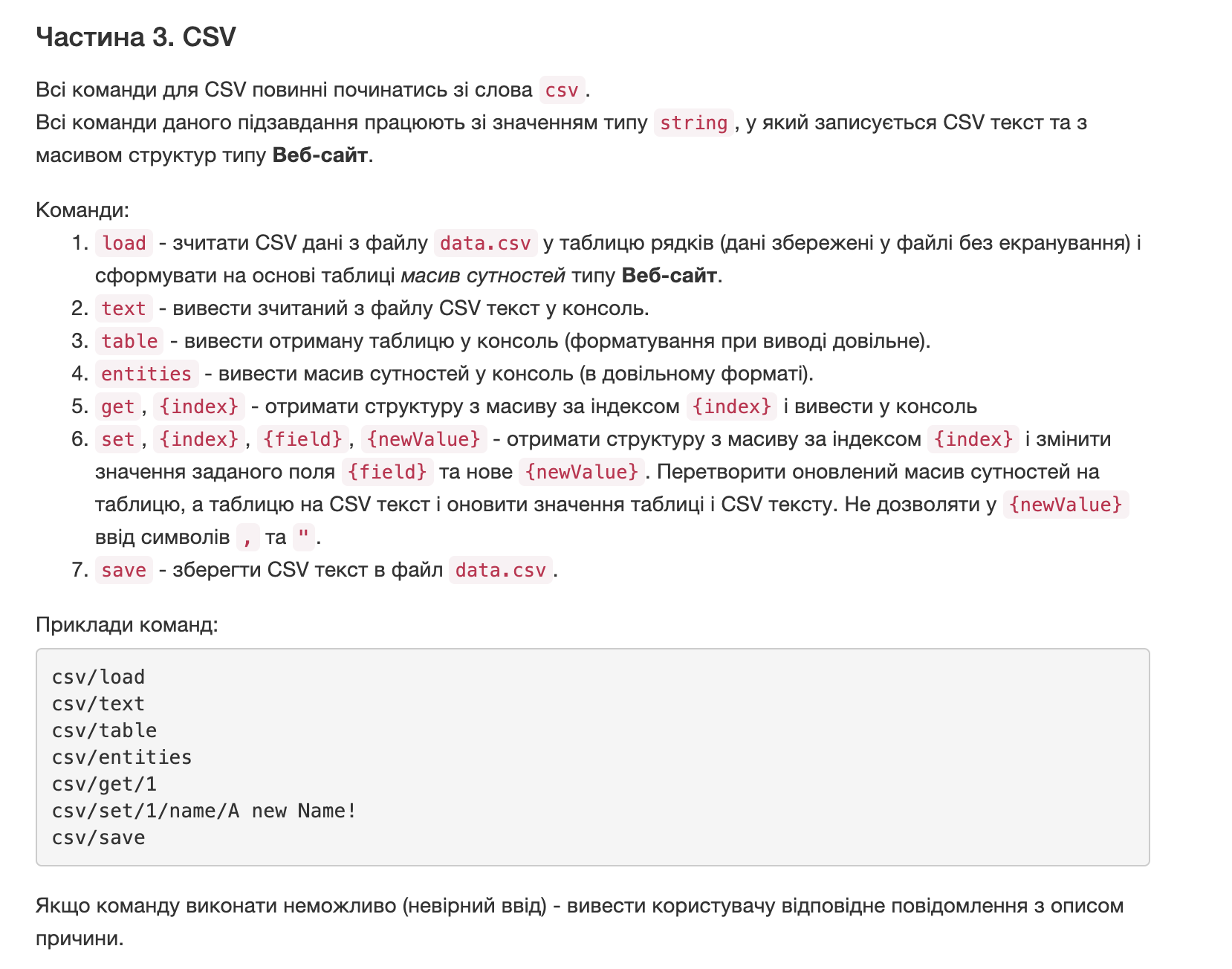
Навчитися працювати з файловими потоками даних для зчитування і зберігання даних.

Використати формат даних CSV для зберігання даних програми на файловій системі.

Завдання





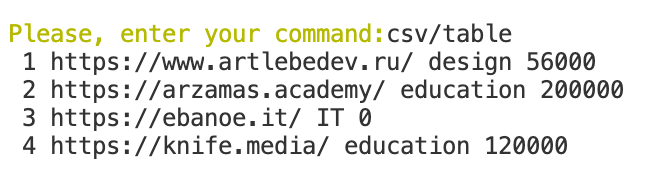


**Аналіз вимог і проектування**

* **CSV текст з даними сутностей за варіантом**

|  |  |  |  |
| --- | --- | --- | --- |
| id | address | topic | number of visitors |
| 1 | <https://www.artlebedev.ru/> | design | 56000 |
| 2 | <https://arzamas.academy/> | education | 200000 |
| 3 | <https://ebanoe.it/> | IT | 0 |
| 4 | <https://knife.media/> | education | 120000 |

* таблицю рядків даних (розібраний CSV).таблицю рядків даних (розібраний CSV).



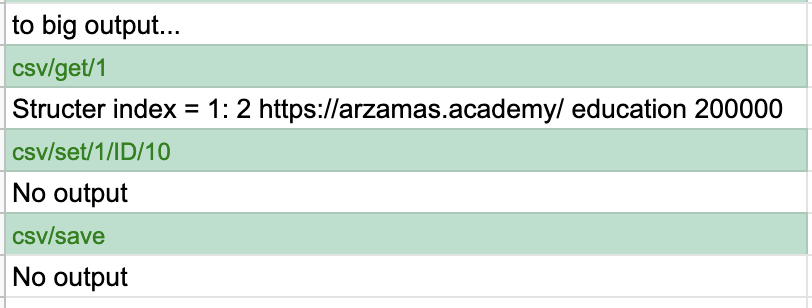
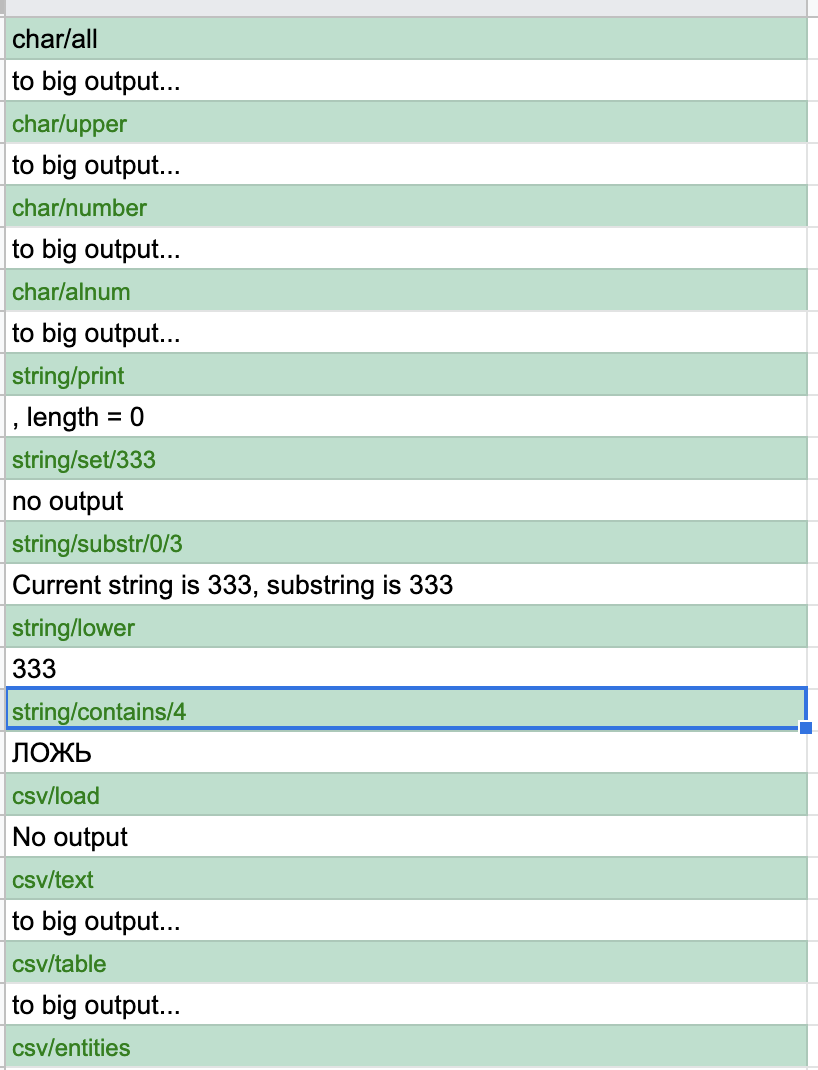
* текстовий опис вашої структури даних (сутності) за варіантом

Структура даних Site. Перше поле - ID, друге - адреса сайту address, наступне - topic, яка тематика сайту. Останнє - numberOfVisitors, скільки людей відвідує сайт.

Тексти коду програм

|  |
| --- |
| **program.cs** |
| using System;  using static System.Console;  using static System.IO.File;  namespace lab5  {  struct Site//struct for part 3  {using System;  using static System.Console;  using static System.IO.File;  namespace lab5  {  struct Site//struct for part 3  {  public int ID;  public string addres;  public string topic;  public int numberOfVisitors;  }  class Program  {  /\*  char/all  char/upper  char/number  char/alnum  string/print  string/set/{A new string}  string/substr/{start index}/{length}  string/lower  string/contains/{char}  csv/load  csv/text  csv/table  csv/entities  csv/get/1  csv/set/1/name/A new Name!  csv/save  \*/  static void Main(string[] args)  {  Console.Clear();//accurate interface  Console.ForegroundColor = ConsoleColor.Blue;  WriteLine("Welcome!");  Console.ResetColor();  while(true)//main part  {  Console.ForegroundColor = ConsoleColor.Yellow;  Write("Please, enter your command:");  Console.ResetColor();  string command = ReadLine();//variable for saving commands  if(command == "Exit")//command for exit from program  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("I WARNED YOU...");  Console.ResetColor();  break;  }  string[] subCommands = command.Split("/");//splitting command  if(subCommands[0] == "char")//part for char commands  {  ProcessChar(subCommands);  }  else if(subCommands[0] == "string")//part for string commands  {  ProcessString(subCommands);  }  else if (subCommands[0] == "csv")//part for CSV commands  {  ProcessCsv(subCommands);  }  else//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Your command is {0}, but there is no such command...", command);  Console.ResetColor();  }  }  }  static void ProcessChar(string[] subCommands)//for all character commands  {  if(subCommands.Length != 2)//checking is input correct  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  return;  }  if(subCommands[1] == "all")  {  PrintAllChar();  }  else if(subCommands[1]== "upper")  {  PrintUpperChar();  }  else if(subCommands[1] == "number")  {  PrintNumberChar();  }  else if(subCommands[1] == "alnum")  {  PrintAlnumChar();  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Your subcommand is |{0}|, its wrong!", subCommands[1]);  Console.ResetColor();  }  }  static void PrintAllChar()  {  for(int i = 0; i <= 127; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintUpperChar()  {  for(int i = 65; i <= 90; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintNumberChar()  {  for(int i = 48; i <= 57; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintAlnumChar()  {  PrintNumberChar();  PrintUpperChar();  for(int i = 97; i <= 122; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static string currentString = "";  static void ProcessString(string[] subCommands)//for all string commands  {  if(subCommands.Length < 2 || subCommands.Length > 4)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be lower than 5 and greater than 1", subCommands.Length);  Console.ResetColor();  return;  }  if(subCommands[1] == "print")  {  if(subCommands.Length != 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  }  else  {  PrintCurrentString(currentString);  }  }  else if(subCommands[1] == "set")  {  if(subCommands.Length != 3)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 3", subCommands.Length);  Console.ResetColor();  }  else  {  SetNewString(ref currentString, subCommands[2]);  }  }  else if(subCommands[1] == "substr")  {  if(subCommands.Length != 4)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 4", subCommands.Length);  Console.ResetColor();  }  else  {  PrintSubString(currentString, subCommands[2], subCommands[3]);  }  }  else if(subCommands[1] == "lower")  {  if(subCommands.Length != 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  }  else  {  PrintInLowerCase(currentString);  }  }  else if( subCommands[1] == "contains")  {  if(subCommands.Length != 3)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 3", subCommands.Length);  Console.ResetColor();  }  else  {  IsStringContainsChar(currentString, subCommands);  }  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like there is no such command", subCommands.Length);  Console.ResetColor();  }  }  static void PrintCurrentString(string currentString)  {  WriteLine("{0}, length = {1}", currentString, currentString.Length);  }  static void SetNewString(ref string currentString, string newString)  {  currentString = newString;  }  static void PrintSubString(string mainString, string startIndex, string length)  {  int startIndexInt;  bool isStartIndexInt = int.TryParse(startIndex, out startIndexInt);  int lengthInt;  bool isLengthInt = int.TryParse(length, out lengthInt);  if(isStartIndexInt == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", startIndex);  Console.ResetColor();  }  else if(startIndexInt < 0)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Start Index = {0}, but it cannot be negative", startIndex);  Console.ResetColor();  }  else if(isLengthInt == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", length);  Console.ResetColor();  }  else if(lengthInt < 0)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Length = {0}, but it cannot be negative", length);  Console.ResetColor();  }  else if(lengthInt + startIndexInt > mainString.Length)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Check your input of lenght and start index", length);  Console.ResetColor();  }  else  {  WriteLine("Current string is {0}, substring is {1}",currentString, mainString.Substring(startIndexInt, lengthInt));  }  }  static void PrintInLowerCase(string currentString)  {  WriteLine(currentString.ToLower());  }  static void IsStringContainsChar(string currentString, string[] subCommands)  {  char character;  bool IsCharacter = char.TryParse(subCommands[2], out character);  if(IsCharacter == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not a character", subCommands[2]);  Console.ResetColor();  }  else  {  if(currentString.Contains(character))  {  WriteLine("True");  }  else  {  WriteLine("False");  }  }  }  static string CsvText = "";//part 3  static string[,] Table = new string[0,0];  static Site[] Sites = new Site[0];  static void ProcessCsv(string[] subCommands)  {  if(subCommands[1] == "load")  {  if(subCommands.Length > 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  LoadData();  }  }  else if(subCommands[1] == "text")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintCsvText();  }  }  else if(subCommands[1] == "table")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintTable();  }  }  else if(subCommands[1] == "entities")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintSites();  }  }  else if(subCommands[1] == "get")  {  if(subCommands.Length != 3)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  return;  }  int IndexInt;  bool IsInt = int.TryParse(subCommands[2], out IndexInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int");  Console.ResetColor();  }  else  {  GetSyte(IndexInt);  }  }  else if(subCommands[1] == "set")  {  if(subCommands.Length != 5)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  return;  }  int IndexInt;  bool IsInt = int.TryParse(subCommands[2], out IndexInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", subCommands[2]);  Console.ResetColor();  }  else if(subCommands[3] != "ID" && subCommands[3] != "address" && subCommands[3] != "topic" && subCommands[3] != "numberOfVisitors")  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such field");  Console.ResetColor();  }  else  {  ProcessSet(IndexInt, subCommands[3], subCommands[4]);  }  }  else if(subCommands[1] == "save")  {  Save();  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  }  static string[,] FormatingCsvToTable(string CsvText)  {  string[] arrayOfRaws = CsvText.Split("\n");  string[] singleRaw = arrayOfRaws[1].Split(",");  Table = new string[arrayOfRaws.Length, singleRaw.Length];  for(int i = 1; i < 5; i++)  {  string[] currentRaw = arrayOfRaws[i].Split(",");  for(int j = 0; j < currentRaw.Length; j++)  {  Table[i, j] = currentRaw[j];  }  }  return Table;  }  static Site[] FormatingTableToSites(string[,] Table)  {  Sites = new Site[Table.GetLength(0)];  for(int i = 1; i < Table.GetLength(0); i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  if(j == 0)  {  Sites[i].ID = int.Parse(Table[i, j]);  }  if(j == 1)  {  Sites[i].addres = Table[i, j];  }  if(j == 2)  {  Sites[i].topic = Table[i, j];  }  if(j == 3)  {  Sites[i].numberOfVisitors = int.Parse(Table[i, j]);  }  }  }  return Sites;  }  static void LoadData()  {  CsvText = ReadAllText("./data.csv");  Table = FormatingCsvToTable(CsvText);  Sites = FormatingTableToSites(Table);  }  static void PrintCsvText()  {  WriteLine(CsvText);  }  static void PrintTable()  {  for(int i = 1; i < Table.GetLength(0); i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  Write(" {0}", Table[i, j]);  }  WriteLine();  }  }  static void PrintSites()  {  for(int i = 1; i < Sites.Length; i++)  {  Write("Structure number {0}: {1} {2} {3} {4} ", i, Sites[i].ID, Sites[i].addres, Sites[i].topic, Sites[i].numberOfVisitors);  WriteLine();  }  }  static void GetSyte(int index)  {  WriteLine("Structer index = {0}: {1} {2} {3} {4}", index, Sites[index + 1].ID, Sites[index + 1].addres, Sites[index + 1].topic, Sites[index + 1].numberOfVisitors);  }  static void ProcessSet(int index, string field, string newField)  {  if(newField.Contains(","))  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Error. New field contains banned symbol");  Console.ResetColor();  return;  }  if(field == "ID" || field == "numberOfVisitors")  {  int newFieldInt;  bool IsInt = int.TryParse(newField, out newFieldInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", newField);  Console.ResetColor();  }  else  {  if(index >= Sites.Length - 1 || index < 0)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Wrong index");  Console.ResetColor();  return;  }  if(field == "ID")  {  Sites[index + 1].ID = newFieldInt;  }  else  {  Sites[index + 1].numberOfVisitors = newFieldInt;  }  }  }  else if(field == "address")  {  Sites[index + 1].addres = newField;  }  else if(field == "topic")  {  Sites[index + 1].topic = newField;  }  Table = ConvertSitesToTable(Sites);  CsvText = ConvertTableToText(Table);  }  static string[,] ConvertSitesToTable(Site[] Sites)  {  for(int i = 1; i < Sites.Length; i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  if(j == 0)  {  Table[i, j] = Convert.ToString(Sites[i].ID);  }  if(j == 1)  {  Table[i, j] = Sites[i].addres;  }  if(j == 2)  {  Table[i, j] = Sites[i].topic;  }  if(j == 3)  {  Table[i, j] = Convert.ToString(Sites[i].numberOfVisitors);  }  }  }  return Table;  }  static string ConvertTableToText(string[,] Table)  {  string[] raws = new string[Table.GetLength(0)];  for(int i = 1; i < Table.GetLength(0); i++)  {  string[] raws1 = new string[Table.GetLength(1)];  for(int j = 0; j < Table.GetLength(1); j++)  {  raws1[j] = Table[i,j];  }  string currentString = string.Join(",", raws1);  raws[i] = currentString;  }  raws[0] = "id,address,topic,number of visitors";  CsvText = string.Join("\r\n", raws);  return CsvText;  }  static void Save()  {  WriteAllText("./data.csv", CsvText);  }  }  }  public int ID;  public string addres;  public string topic;  public int numberOfVisitors;  }  class Program  {  /\*  char/all  char/upper  char/number  char/alnum  string/print  string/set/{A new string}  string/substr/{start index}/{length}  string/lower  string/contains/{char}  csv/load  csv/text  csv/table  csv/entities  csv/get/1  csv/set/1/name/A new Name!  csv/save  \*/  static void Main(string[] args)  {  Console.Clear();//accurate interface  Console.ForegroundColor = ConsoleColor.Blue;  WriteLine("Welcome!");  Console.ResetColor();  while(true)//main part  {  WriteLine();  Console.ForegroundColor = ConsoleColor.Green;  WriteLine("Here is the list of avaliable commands and short explanation:");  WriteLine("char/all - return all ASCII characters");  WriteLine("char/upper - return all ASCII characters that are in upper case");  WriteLine("char/number - return all ASCII numbers");  WriteLine("char/alnum - return all ASCII letters and digits");  WriteLine("string/print - print current string and its length;");  WriteLine("string/set/{A new string} - change value of current string to new string");  WriteLine("string/substr/{start index}/{length} - print substring from curremnt string");  WriteLine("string/lower - print current string in lower case");  WriteLine("string/contains/{char} - check if current ctring contains char");  WriteLine("Exit - you dont need to input this command, trust me, it's the best program you've ever seen!");  Console.ResetColor();  WriteLine();  Console.ForegroundColor = ConsoleColor.Yellow;  Write("Please, enter your command:");  Console.ResetColor();  string command = ReadLine();//variable for saving commands  if(command == "Exit")//command for exit from program  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("I WARNED YOU...");  Console.ResetColor();  break;  }  string[] subCommands = command.Split("/");//splitting command  if(subCommands[0] == "char")//part for char commands  {  ProcessChar(subCommands);  }  else if(subCommands[0] == "string")//part for string commands  {  ProcessString(subCommands);  }  else if (subCommands[0] == "csv")//part for CSV commands  {  ProcessCsv(subCommands);  }  else//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Your command is {0}, but there is no such command...", command);  Console.ResetColor();  }  }  }  static void ProcessChar(string[] subCommands)//for all character commands  {  if(subCommands.Length != 2)//checking is input correct  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  return;  }  if(subCommands[1] == "all")  {  PrintAllChar();  }  else if(subCommands[1]== "upper")  {  PrintUpperChar();  }  else if(subCommands[1] == "number")  {  PrintNumberChar();  }  else if(subCommands[1] == "alnum")  {  PrintAlnumChar();  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Your subcommand is |{0}|, its wrong!", subCommands[1]);  Console.ResetColor();  }  }  static void PrintAllChar()  {  for(int i = 0; i <= 127; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintUpperChar()  {  for(int i = 65; i <= 90; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintNumberChar()  {  for(int i = 48; i <= 57; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static void PrintAlnumChar()  {  PrintNumberChar();  PrintUpperChar();  for(int i = 97; i <= 122; i++)  {  WriteLine("ASCII code:{0}, character:|{1}|", i, (char)i);  }  }  static string currentString = "";  static void ProcessString(string[] subCommands)//for all string commands  {  if(subCommands.Length < 2 || subCommands.Length > 4)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be lower than 5 and greater than 1", subCommands.Length);  Console.ResetColor();  return;  }  if(subCommands[1] == "print")  {  if(subCommands.Length != 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  }  else  {  PrintCurrentString(currentString);  }  }  else if(subCommands[1] == "set")  {  if(subCommands.Length != 3)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 3", subCommands.Length);  Console.ResetColor();  }  else  {  SetNewString(ref currentString, subCommands[2]);  }  }  else if(subCommands[1] == "substr")  {  if(subCommands.Length != 4)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 4", subCommands.Length);  Console.ResetColor();  }  else  {  PrintSubString(currentString, subCommands[2], subCommands[3]);  }  }  else if(subCommands[1] == "lower")  {  if(subCommands.Length != 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 2", subCommands.Length);  Console.ResetColor();  }  else  {  PrintInLowerCase(currentString);  }  }  else if( subCommands[1] == "contains")  {  if(subCommands.Length != 3)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like your command length is {0}, but should be 3", subCommands.Length);  Console.ResetColor();  }  else  {  IsStringContainsChar(currentString, subCommands);  }  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Looks like there is no such command", subCommands.Length);  Console.ResetColor();  }  }  static void PrintCurrentString(string currentString)  {  WriteLine("{0}, length = {1}", currentString, currentString.Length);  }  static void SetNewString(ref string currentString, string newString)  {  currentString = newString;  }  static void PrintSubString(string mainString, string startIndex, string length)  {  int startIndexInt;  bool isStartIndexInt = int.TryParse(startIndex, out startIndexInt);  int lengthInt;  bool isLengthInt = int.TryParse(length, out lengthInt);  if(isStartIndexInt == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", startIndex);  Console.ResetColor();  }  else if(startIndexInt < 0)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Start Index = {0}, but it cannot be negative", startIndex);  Console.ResetColor();  }  else if(isLengthInt == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", length);  Console.ResetColor();  }  else if(lengthInt < 0)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Length = {0}, but it cannot be negative", length);  Console.ResetColor();  }  else if(lengthInt + startIndexInt > mainString.Length)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Check your input of lenght and start index", length);  Console.ResetColor();  }  else  {  WriteLine("Current string is {0}, substring is {1}",currentString, mainString.Substring(startIndexInt, lengthInt));  }  }  static void PrintInLowerCase(string currentString)  {  WriteLine(currentString.ToLower());  }  static void IsStringContainsChar(string currentString, string[] subCommands)  {  char character;  bool IsCharacter = char.TryParse(subCommands[2], out character);  if(IsCharacter == false)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not a character", subCommands[2]);  Console.ResetColor();  }  else  {  if(currentString.Contains(character))  {  WriteLine("True");  }  else  {  WriteLine("False");  }  }  }  static string CsvText = "";//part 3  static string[,] Table = new string[0,0];  static Site[] Sites = new Site[0];  static void ProcessCsv(string[] subCommands)  {  if(subCommands[1] == "load")  {  if(subCommands.Length > 2)//checking input  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  LoadData();  }  }  else if(subCommands[1] == "text")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintCsvText();  }  }  else if(subCommands[1] == "table")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintTable();  }  }  else if(subCommands[1] == "entities")  {  if(subCommands.Length > 2)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  else  {  PrintSites();  }  }  else if(subCommands[1] == "get")  {  if(subCommands.Length != 3)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  return;  }  int IndexInt;  bool IsInt = int.TryParse(subCommands[2], out IndexInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int");  Console.ResetColor();  }  else  {  GetSyte(IndexInt);  }  }  else if(subCommands[1] == "set")  {  if(subCommands.Length != 5)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  return;  }  int IndexInt;  bool IsInt = int.TryParse(subCommands[2], out IndexInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", subCommands[2]);  Console.ResetColor();  }  else if(subCommands[3] != "ID" && subCommands[3] != "address" && subCommands[3] != "topic" && subCommands[3] != "numberOfVisitors")  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such field");  Console.ResetColor();  }  else  {  ProcessSet(IndexInt, subCommands[3], subCommands[4]);  }  }  else if(subCommands[1] == "save")  {  Save();  }  else  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. There is no such command");  Console.ResetColor();  }  }  static string[,] FormatingCsvToTable(string CsvText)  {  string[] arrayOfRaws = CsvText.Split("\n");  string[] singleRaw = arrayOfRaws[1].Split(",");  Table = new string[arrayOfRaws.Length, singleRaw.Length];  for(int i = 1; i < 5; i++)  {  string[] currentRaw = arrayOfRaws[i].Split(",");  for(int j = 0; j < currentRaw.Length; j++)  {  Table[i, j] = currentRaw[j];  }  }  return Table;  }  static Site[] FormatingTableToSites(string[,] Table)  {  Sites = new Site[Table.GetLength(0)];  for(int i = 1; i < Table.GetLength(0); i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  if(j == 0)  {  Sites[i].ID = int.Parse(Table[i, j]);  }  if(j == 1)  {  Sites[i].addres = Table[i, j];  }  if(j == 2)  {  Sites[i].topic = Table[i, j];  }  if(j == 3)  {  Sites[i].numberOfVisitors = int.Parse(Table[i, j]);  }  }  }  return Sites;  }  static void LoadData()  {  CsvText = ReadAllText("./data.csv");  Table = FormatingCsvToTable(CsvText);  Sites = FormatingTableToSites(Table);  }  static void PrintCsvText()  {  WriteLine(CsvText);  }  static void PrintTable()  {  for(int i = 1; i < Table.GetLength(0); i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  Write(" {0}", Table[i, j]);  }  WriteLine();  }  }  static void PrintSites()  {  for(int i = 1; i < Sites.Length; i++)  {  Write("Structure number {0}: {1} {2} {3} {4} ", i, Sites[i].ID, Sites[i].addres, Sites[i].topic, Sites[i].numberOfVisitors);  WriteLine();  }  }  static void GetSyte(int index)  {  WriteLine("Structer index = {0}: {1} {2} {3} {4}", index, Sites[index + 1].ID, Sites[index + 1].addres, Sites[index + 1].topic, Sites[index + 1].numberOfVisitors);  }  static void ProcessSet(int index, string field, string newField)  {  if(newField.Contains(","))  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Error. New field contains banned symbol");  Console.ResetColor();  return;  }  if(field == "ID" || field == "numberOfVisitors")  {  int newFieldInt;  bool IsInt = int.TryParse(newField, out newFieldInt);  if(IsInt == false)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. {0} is not an int", newField);  Console.ResetColor();  }  else  {  if(index >= Sites.Length - 1 || index < 0)  {  Console.ForegroundColor = ConsoleColor.Red;  WriteLine("Hey, somethig went wrong. Wrong index");  Console.ResetColor();  return;  }  if(field == "ID")  {  Sites[index + 1].ID = newFieldInt;  }  else  {  Sites[index + 1].numberOfVisitors = newFieldInt;  }  }  }  else if(field == "address")  {  Sites[index + 1].addres = newField;  }  else if(field == "topic")  {  Sites[index + 1].topic = newField;  }  Table = ConvertSitesToTable(Sites);  CsvText = ConvertTableToText(Table);  }  static string[,] ConvertSitesToTable(Site[] Sites)  {  for(int i = 1; i < Sites.Length; i++)  {  for(int j = 0; j < Table.GetLength(1); j++)  {  if(j == 0)  {  Table[i, j] = Convert.ToString(Sites[i].ID);  }  if(j == 1)  {  Table[i, j] = Sites[i].addres;  }  if(j == 2)  {  Table[i, j] = Sites[i].topic;  }  if(j == 3)  {  Table[i, j] = Convert.ToString(Sites[i].numberOfVisitors);  }  }  }  return Table;  }  static string ConvertTableToText(string[,] Table)  {  string[] raws = new string[Table.GetLength(0)];  for(int i = 1; i < Table.GetLength(0); i++)  {  string[] raws1 = new string[Table.GetLength(1)];  for(int j = 0; j < Table.GetLength(1); j++)  {  raws1[j] = Table[i,j];  }  string currentString = string.Join(",", raws1);  raws[i] = currentString;  }  raws[0] = "id,address,topic,number of visitors";  CsvText = string.Join("\r\n", raws);  return CsvText;  }  static void Save()  {  WriteAllText("./data.csv", CsvText);  }  }  } |

**Приклади результатів та команди**

****

**Приклади результатів при некорректних данних**

|  |
| --- |
| scv//// |
| Your command is scv////, but there is no such command... |
| csv/set/7/ID/10 |
| Hey, somethig went wrong. Wrong index |
| csv/set/333/ffg/ggd |
| Hey, somethig went wrong. There is no such field |

**Висновки**

Під час виконання лабораторної роботи я навчився працювати з рядками символів.

Вмію визначати класи символів та використовувати функції для роботи з символами.

Навчився використовувати стандартні функції для роботи з рядками символів.

Навчився працювати з файловими потоками даних для зчитування і зберігання даних.

Використав формат даних CSV для зберігання даних програми на файловій системі.