

ДЕНЬ 2С. Работа с файлами

2С.1. Прочитать данные из консоли и сформировать CSV-файл

Пользователь вводит записи, состоящие из нескольких полей. Как только он вводит пустую строку в первое из полей, ввод завершается и пользователю предлагается ввести путь до файла для сохранения. Необходимо сохранить записи в виде CSV-файла (табличные данные, в которых каждая строка — это строка таблицы, а колонки разделены запятыми или точкой с запятой) по указанному пути

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число).

Программа:

Введите данные:

Фамилия: Петров
Возраст: 20
Результат: 5.3

Фамилия: Иванов
Возраст: 22
Результат: 7.6

Фамилия: Сидоров
Возраст: 20
Результат: 5

Фамилия:

Введите путь для сохранения файла:
> data_new.csv

Выходной файл:

Петров, 20, 5.3
Иванов, 22, 7.6
Сидоров, 20, 5

Варианты

№	Запись
1	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)
2	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)
3	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)
4	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)
5	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)
6	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)
7	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)

8	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)
9	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)
10	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)
11	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)
12	Дисциплина (строка), номер курса (целое число), количество часов (целое число)
13	Название материала (строка), объем (вещественное число), вес (вещественное число)
14	Фамилия (строка), должность (строка), оклад в руб (целое число)
15	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)
16	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)
17	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)
18	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)
19	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)
20	Фамилия (строка), рост (вещественное число), вес (вещественное число)

2С.2. Калькулятор площадей участков считывающий данные из файла

Необходимо разработать калькулятор площадей участков разного размера. Формула площади участка и число вводимых параметров зависят от варианта.

При запуске программа спрашивает пользователя путь до CSV-файла с перечнем участков. В CSV-файле первой строкой идет заголовок таблицы, по которому определяется порядок колонок в файле. Например, если участок задается тремя параметрами a , b и c , а первая строка в CSV-файле: b , c , a — то это означает, что первая колонка – параметр b , вторая – параметр c , третья – параметр a .

После ввода, программа должна вывести площади каждого из участков и сводный показатель согласно варианту.

Пример

Формула площади: $a * b$, параметры: a и b , сводный показатель: сумма площадей

Входной файл *grounds.csv*:

b , a
 10, 3
 2.5, 5
 8, 5

Программа:

Введите путь до файла с участками
> grounds.csv

- 1) 30
- 2) 12.5
- 3) 40

Сумма площадей: 72.5

Варианты

№	Формула площади	Параметры	Сводный показатель
1	$a * b * \sin(c) / 2$	a, b, c	Минимальная площадь
2	$a * h / 2$	a, h	Максимальная площадь
3	$a * b$	a, b	Максимальная площадь
4	$a * h / 2$	a, h	Минимальная площадь
5	$h * (a + b) / 2$	a, b, h	Минимальная площадь
6	$a * h / 2$	a, h	Сумма площадей
7	$a * b$	a, b	Сумма площадей
8	$r * (a + b + c) / 2$	a, b, c, r	Разброс площадей (разница между максимальной и минимальной площадями)
9	$a * b * \sin(c) / 2$	a, b, c	Максимальная площадь
10	$a * a * \tan(p) / 2$	a, p	Разброс площадей (разница между максимальной и минимальной площадями)
11	$h * (a + b) / 2$	a, b, h	Разброс площадей (разница между максимальной и минимальной площадями)
12	$h * (a + b) / 2$	a, b, h	Сумма площадей
13	$a * b * \sin(c) / 2$	a, b, c	Сумма площадей
14	$r * (a + b + c) / 2$	a, b, c, r	Максимальная площадь
15	$a * b$	a, b	Минимальная площадь
16	$a * a * \tan(p) / 2$	a, p	Максимальная площадь
17	$r * (a + b + c) / 2$	a, b, c, r	Сумма площадей
18	$a * h / 2$	a, h	Разброс площадей (разница между максимальной и минимальной площадями)
19	$a * b * \sin(c) / 2$	a, b, c	Разброс площадей (разница между максимальной и минимальной площадями)
20	$a * a * \tan(p) / 2$	a, p	Минимальная площадь

2С.3. Прочитать все файлы из указанной директории и сформировать объединенный CSV-файл с учетом фильтра

Пользователь указывает путь до директории с исходными CSV файлами и путь до файла с результатом. Необходимо прочитать все исходные CSV-файлы в директории и записать в результирующий CSV-файл только строки, попадающие под заданное условие. В выходном файле помимо колонок исходного файла должна быть добавлена колонка с названием файла-источника.

Для получения списка файлов, см. *System.IO.Directory.GetFiles*

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число).
Вывести записи, у которых возраст меньше 21

Входной файл jan.csv:

Петров, 20, 5.3
Иванов, 22, 7.6
Сидоров, 20, 5

Входной файл feb.csv:

Потемкин, 19, 2.3
Пушкин, 32, 9.6

Входной файл mar.csv:

Журавлев, 18, 7.3
Никитин, 33, 8.6
Жаров, 22, 3.5
Семечкин, 55, 4

Программа:

Введите путь к директории:
> data/dir/

Введите путь для сохранения файла:
> report.csv

Выходной файл:

jan.csv, Петров, 20, 5.3
jan.csv, Сидоров, 20, 5
feb.csv, Потемкин, 19, 2.3
mar.csv, Журавлев, 18, 7.3

Варианты

№	Запись	Условие
1	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)	Номер этажа > 2
2	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)	Вес > 500
3	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)	Число выполненных заданий >= 7
4	Название материала (строка), объем (вещественное число), вес (вещественное число)	Объем < 10
5	Фамилия (строка), должность (строка), оклад в руб (целое число)	Оклад > 25000

6	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)	Число уникальных посетителей > 1000
7	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)	Значение плана выпуска деталей больше значения фактического выпуска деталей
8	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)	Пробег < 10000
9	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)	Средняя цена > 100
10	Фамилия (строка), рост (вещественное число), вес (вещественное число)	Рост > 200
11	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)	Число слов > 200
12	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)	Средний балл >= 4
13	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)	Число сезонов > 2
14	Дисциплина (строка), номер курса (целое число), количество часов (целое число)	Количество часов < 40
15	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)	Разница между суммами поступления и списания > 0
16	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)	Баланс < 0
17	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)	Среднее между оценками за теорию и практику >= 3.5
18	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)	Сумма заказа > 800
19	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)	Количество на складе < количества зарезервированных
20	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)	Гонорар >= 1

2С.4. Обработка файла с учетом аргументов командной строки

Вам необходимо реализовать программу, принимающую входную информацию через аргументы командной строки.

Пример запуска программы:

```
program.exe input.txt output.txt -from 5 -to 8
```

Аргументы:

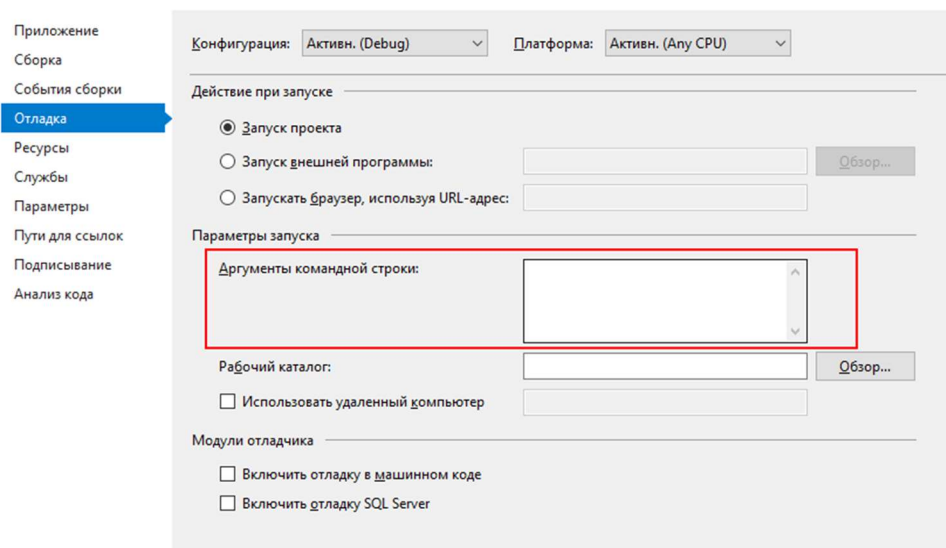
- *input.txt* – входной файл, который нужно обработать. Обязательный аргумент. Если не задан, то выдать сообщение «Укажите входной файл» и завершить работу
- *output.txt* – выходной файл, куда нужно сохранить результат. Если аргумент не передан, то результат нужно вывести в консоль. Этот аргумент всегда идет после аргумента с входным файлом
- *-from 5* – с какой строки файла начинать обработку. Если не задан, то обработка начинается с начала файла
- *-to 8* – до какой строки файла, включительно, проводить обработку. Если не задан, то обработка производится до конца файла

Порядок аргументов может быть произвольным

Получить значение аргументов можно в главной функции программы:

```
static void Main(string[] args) // args - аргументы
```

Для тестирования указать аргументы командной строки можно в настройках проекта:



Пример

Обработка: удаление каждого второго слова

Файл *poem.txt*

В томленьях грусти безнадежной
В тревогах шумной суеты,
Звучал мне долго голос нежный
И снились милые черты.

Шли годы. Бурь порыв мятежный
Рассеял прежние мечты,
И я забыл твой голос нежный,
Твои небесные черты.

Запуск программы:

```
program.exe -from 2 poem.txt res.txt -to 6
```

После запуска программы должен создаться *res.txt*:

В шумной,
Звучал долго нежный
И милые

Шли Бурь мятежный

Варианты

№	Преобразование	№	Преобразование
1	Удалить знаки препинания	11	Перевести первую букву каждого слова в верхний регистр
2	Удалить все согласные	12	Удалить все слова без согласных букв
3	Перевести в верхний регистр все гласные	13	Удаление каждого второго слова
4	Удалить все слова, в которых меньше 5 согласных	14	Перевести в верхний регистр все согласные
5	Перевести в верхний регистр каждую вторую букву слов: абв гдеж -> аБв гДеЖ	15	Удалить первое и последнее слова в строках
6	Удалить все слова размером больше, чем из 5 букв	16	Перемешать в случайном порядке все слова в строке
7	Поменять местами слова в каждой паре слов: аб вг де жз -> вг аб жз де	17	Перевести все буквы первого слова каждой строке в верхний регистр
8	Удалить все слова, в которых больше 5 согласных	18	Перевернуть строки (абг деж -> жед гба)
9	Удалить все слова размером меньше, чем из 5 букв	19	Перевернуть каждое слово в строке (абг деж -> гба жед)
10	Удалить все слова без гласных букв	20	Удалить все гласные

2С.5. Отобразить BMP-файл с изображением в консоли

Пользователь указывает путь до файла с изображением размером от 8x8 до 24x24. Необходимо открыть, выполнить заданное преобразование, а затем вывести изображение в консоли, используя заданные тона. Каждый пиксел изображения при выводе должен состоять из двух пробельных символов. Цвет «пикселя» должен задаваться с помощью свойства *Console.BackgroundColor*

Для открытия и работы с изображением используйте класс *System.Drawing.Bitmap*. Подключите в проекте ссылку на стандартную библиотеку *System.Drawing.dll*, если класс недоступен.

Для получения цвета пикселя используйте *Bitmap.GetPixel(Int32, Int32)*. Для выбора отображаемого цвета, вычислите среднее значение из каналов R,G,B (яркость):

- $0 \leq \text{Яркость} < 60$ – цвет: Black
- $60 \leq \text{Яркость} < 120$ – цвет: <Цвет2>
- $120 \leq \text{Яркость} < 180$ – цвет: <Цвет1>
- $180 \leq \text{Яркость}$ – цвет: White

Пример

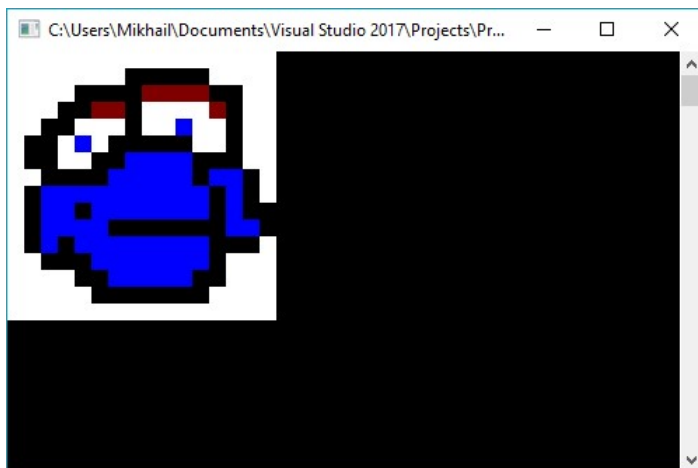
Входное изображение (16x16): 🍷

Цвет1: Blue

Цвет2: DarkRed

Преобразование: Отражение по вертикали

Результат:



Варианты

№	Цвет 1	Цвет 2	Преобразование
1	Green	DarkMagenta	Уменьшение размера в два раза
2	Red	DarkGray	Увеличение размера в два раза
3	Blue	DarkCyan	Отражение по вертикали
4	Magenta	DarkGreen	Уменьшение размера в два раза
5	Green	DarkMagenta	Поворот на 180°
6	Cyan	DarkBlue	Увеличение размера в два раза
7	Gray	DarkCyan	Отражение по горизонтали
8	Magenta	DarkGreen	Обрезка 2 пикселей по периметру
9	Green	DarkYellow	Обрезка 2 пикселей по периметру
10	Yellow	DarkYellow	Поворот на 180°
11	Magenta	DarkYellow	Обрезка 2 пикселей по периметру
12	Blue	DarkYellow	Отражение по вертикали
13	Gray	DarkMagenta	Отражение по горизонтали
14	Cyan	DarkGreen	Отражение по горизонтали
15	Red	DarkCyan	Увеличение размера в два раза
16	Yellow	DarkGray	Отражение по горизонтали
17	Blue	DarkRed	Отражение по вертикали
18	Cyan	DarkBlue	Уменьшение размера в два раза
19	Red	DarkGray	Поворот на 180°
20	Gray	DarkRed	Отражение по вертикали