



**Resource Development**

# **ТЕСТОВОЕ ЗАДАНИЕ**

---

## СОДЕРЖАНИЕ

1	ОБЩЕЕ ОПИСАНИЕ ЗАДАНИЯ.....	3
1.1	ПОИСК ПО CSV ФАЙЛУ .....	3
1.1.1	Общее описание приложения .....	3
1.1.2	Формат файла.....	3
1.1.3	Запуск приложения .....	3
1.1.4	Комплект поставки .....	4
1.2	ГЕНЕРАТОР ТЕСТОВЫХ ДАННЫХ .....	4
1.2.1	Общее описание.....	4
1.2.2	Комплект поставки .....	4
2	ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ .....	4
3	ОБЩИЕ ТРЕБОВАНИЯ К ПОСТАВКЕ .....	5

# 1 ОБЩЕЕ ОПИСАНИЕ ЗАДАНИЯ

Задание состоит из двух взаимосвязанных частей, которые должны быть реализованы в виде отдельных независимых приложений:

- Поиск по CSV файлу
- Генератор тестовых данных

## 1.1 ПОИСК ПО CSV ФАЙЛУ

### 1.1.1 Общее описание приложения

Приложение работает с входным файлом в формате CSV (Comma Separated Values). В файле содержатся табличные данные. Приложение позволяет осуществлять поиск по значению в указанной колонке. Результат поиска (строки, содержащие найденные значения) выводится в выходной файл в формате CSV.

### 1.1.2 Формат файла

CSV – это текстовый формат, предназначенный для представления табличных данных. Его основные характеристики:

- Каждая строка файла – это одна строка таблицы
- Значения отдельных колонок разделяются символом «точка с запятой» (;)
- Значения, содержащие зарезервированные символы, такие как: «точка с запятой» или «новая строка» - обрамляются символом «двойные кавычки» (")
- Если в значении встречаются кавычки – они представляются в файле в виде двух кавычек подряд ("" )
- Строки разделяются парой символов CR LF (0x0D 0x0A).

В первой строке файла через точку с запятой перечислены имена колонок и их типов через пробел. Имена колонок не должны содержать пробелов. Должны быть доступны следующие типы колонок:

- String – строка
- Date – дата (всегда в формате dd.mm.yyyy)
- Integer – целочисленное значение
- Float – значение с плавающей запятой. Разделитель дробной и целой части – запятая (,)

Пример файла:

```
ФИО String; Дата_рождения Date; Кол-во_проектов Integer; Рейтинг Float;
Комментарий String
Иванов Иван Иванович; 18.06.1983; 34; 6,45; "Работал над проектами: ""АВС"";
""КВД"""
```

### 1.1.3 Запуск приложения

Приложение запускается из командной строки:

```
search_csv.exe -in <входной_файл> -out <выходной_файл> -enc <кодировка> -col
<имя_столбца> -expr <выражение_для_поиска>
```

Параметры должны проверяться на корректность, в том числе параметр -expr, который зависит от типа колонки указанной в -col.

Входной файл должен проверяться на корректность формата (в том числе типов данных в колонках)

Пример запуска приложения:

```
search_csv.exe -in in_file.csv -out out_file.csv -enc UTF-8 -col
Дата_рождения -exp 18.06.1983
```

Программа должна успешно обрабатывать файлы, содержащие до 1000 колонок, до 1 000 000 строк, общим объемом до 1 Гб.

#### 1.1.4 Комплект поставки

- Архив search\_bin.zip содержащий бинарные коды.
- Архив search\_src.zip содержащий исходные коды.
- Архив search\_samples.zip содержащий примеры корректных и некорректных файлов в формате csv, охватывающий максимально возможное количество вариантов. Тесты размером более 100 Кб включать не надо.
- Файл README.txt содержащий краткую инструкцию по использованию.

### 1.2 ГЕНЕРАТОР ТЕСТОВЫХ ДАННЫХ

#### 1.2.1 Общее описание

Программа, генерирующая тестовые данные для программы «Поиск по CSV файлу» с заданным количеством колонок и строк.

Пример вызова:

```
gen_csv.exe -col <кол-во_колонок> -row <кол-ко_строк> -len
<макс_длина_значения> -enc <кодировка> -out <имя_выходного_файла>
```

Например:

```
gen_csv.exe -col 543 -row 987654 -len 300 -enc UTF-8 -out in_file.csv
```

В результате выполнения программы по указанному пути должен сформироваться файл в формате CSV (в соответствии с требованиями, изложенными выше). Имена колонок и их типы система должна генерировать случайным образом, а затем формировать строки из случайных символов. Максимальную длину значения типа String можно ограничивать параметром -len.

Программа «Поиск по CSV файлу» должна уметь обрабатывать файлы, сгенерированные данным генератором.

#### 1.2.2 Комплект поставки

- Архив gen\_bin.zip содержащий бинарные коды.
- Архив gen\_src.zip содержащий исходные коды.
- Файл README.txt содержащий краткую инструкцию по использованию.

## 2 ТРЕБОВАНИЯ К РЕАЛИЗАЦИИ

- Приложение с CLI (Command Line Interface) под Win32.
- Язык реализации: C#.
- Обязательная декомпозиция задачи на отдельные классы (использование объектно-ориентированного подхода).
- Использование комментариев ко всем классам, методам и полям на русском языке.
- Парсер и формater CSV необходимо реализовать самостоятельно без использования сторонних библиотек (компонент).
- В случае некорректного использования программы должны выдавать сообщение об ошибке на консоль на русском языке, в котором бы указывалась причина ошибки.

### **3 ОБЩИЕ ТРЕБОВАНИЯ К ПОСТАВКЕ**

- Комплекты поставок каждой подзадачи должны быть упакованы в один архив, имя которого совпадает с вашим именем в транскрипции FirstName\_LastName.zip.
- Общий размер финального архива не должен превышать 1Мб.