

Лабораторная работа №13

Обработка и форматирование строк

1 Цель работы

- 1.1 Изучить процесс обработки строк на языке C#;
- 1.2 Научиться применять стандартные методы классов String, StringBuilder и Char для обработки строковых и символьных данных в программах на языке C#.

2 Литература

- 2.1 Полное руководство по языку программирования C# 11 и платформе .NET 7
<https://metanit.com/sharp/tutorial/> – гл.11.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание лабораторной работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Строки и объекты объявить программно.

5.1 Анализ строки

Написать программу, выводящую на экран:

- количество символов в строке,
- количество символов в строке без учета пробелов,
- количество букв в строке,
- все позиции (индекс), на которых в строке находится указанный пользователем текст (с учетом регистра). Если совпадение не найдено, сообщить об этом.

5.2 Получение и анализ массива строк (по вариантам: 1 - нечетный ПК, 2 - четный ПК)

5.2.1 Написать программу, в которой реализовать разделение строки на массив строк с выводом каждого элемента на экран и сообщением, сколько элементов найдено. Пустых строк и пробелов по краям строк в элементах массива быть не должно. Задание по вариантам:

1) разделить на предложения. Конец предложения определяется символом точки, вопросительный знак, восклицательный знак или их комбинацией.

2) разделить на слова. Слова разделяются пробелами, запятыми, точками или комбинацией этих символов.

5.2.2 Вывести на экран все элементы массива, соответствующие условию без учета регистра (по вариантам). Если совпадение не найдено, сообщить об этом. Условие:

- 1) строки, начинающиеся с указанного пользователем текста,
- 2) строки, содержащие указанный пользователем текст.

5.3 Изменение строки

Написать программу:

- изменяющую строку путем обрезки пробелов по краям строки и замены двух и более подряд идущих пробелов на один. Для выполнения использовать методы String,

- изменяющую регистр букв в строке на указанный пользователем: верхний (ТЕКСТ), нижний (текст), инвертированный (тeКСТ -> ТЕкст). Для посимвольного изменения использовать `StringBuilder`, для приведения к конкретному регистру использовать `String`.

5.4 Форматирование и интерполяция строк (по вариантам из ЛР №5)

Для класса с автосвойствами переопределить и протестировать метод `ToString()` для возврата данных в определенном формате (примеры – в таблице):

- значения должны разделяться точкой с запятой,
- к каждому свойству должен быть применен подходящий формат или спецификатор формата.

Товар: Название, Цена, Срок годности	Формат: название;0.00 р;гггг-мм-дд вместо 0.00 р — цена товара, вместо гггг-мм-дд — срок годности, в названии все буквы строчные.
Пациент: ФИО, Полис, Дата рождения	Формат: ФИО;xxxxxxxxx;гггг/мм/дд вместо xxxxxxxxx — номер полиса (если цифр меньше 9, в начале должны быть нули, дополняющие строку до длины 9 символов, вместо гггг/мм/дд — дата рождения, в ФИО все буквы заглавные.

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio и создать консольное приложение C# (Console Application).

6.2 Выполнить все задания из п.5 в одном решении LabWork13 (для каждого задания создавать внутри решения свой проект Task#, где вместо # указывать номер задания).

При разработке считать, что пользователь ввел данные требуемого типа, остальные возможные ошибки обрабатывать.

При выполнении заданий использовать минимально возможное количество команд и переменных и выполнять форматирование и рефакторинг кода.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 К какому типу переменных относятся переменные типа `string`?
- 8.2 Какие операции допустимы над строковыми данными?
- 8.3 В чем отличие переменной типа `string` от массива символов?
- 8.4 Что такое «интерполяция строк»?
- 8.5 Изменяют ли методы класса `String` исходную строку?
- 8.6 Для чего используется `StringBuilder`?