**Федеральное агентство связи**

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Поволжский государственный университет телекоммуникаций и информатики»**

**Факультет информационных систем и технологий**

**Кафедра программного обеспечения и управления в технических системах**

Лабораторная работа №3

Управление файловой системой в ОС

Выполнил студент

группы МОИС-01

Богрянцев Игорь

Проверил: к.т.н., доцент

Бадыков Р. И.

Самара, 2022 г.

Изображение выглядит как текст

Автоматически созданное описание

Функция создания нового файла:

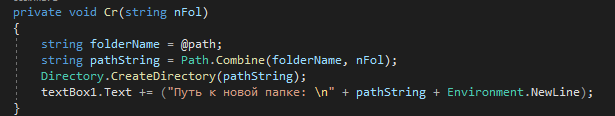
Создаем переменную pathString и присваиваем ему значение Пути и вызываем функцию CreateDirectory и передаем ей аргумент pathString. Выводим значения в текстБокс.

Изображение выглядит как текст

Автоматически созданное описание

Создание папки (директории, каталога):

Передаем функции название файла, после чего, с помощью Combine, создаем конкретную директорию. Выводим информацию в ТекстБокс.



Вывод информации о каталоге:

В функцию Dir передаем путь path, создаем переменные типа int, которые производят подсчет количества файлов и папок в каталоге. Создаем массив объектов типа FileInfo и назовем fileNames. В него передаем файлы со всеми расширениями. Создаем foreach цикл для вывода информации о файлах. Аналогично вышесказанному, создаем объект dir типа DirectoryInfo. С помощью цикла foreach() и метода .Getdirectories, получаем список папок, находящихся по заданному пути. В цикле происходит и проверка на ограничение системы EXT – «Длина имени папки не превышает 255 символов». Затем производится подсчет количества папок с помощью переменной papki++;

После вышеперечисленного, в textBox1 передается строка с нужными нам данными.

Изображение выглядит как текст

Автоматически созданное описание

Копирование директории и файлов:

В функцию CopyDir(), передаются аргументы sourcePath- изначальный путь директории, targetPath-конечный путь копирования файлов. В функции первым делом создается переменная типа string filename и ей присваивается 0 значение. После чего происходит создание пути к файлы с помощью метода Path.Combine(), в который передаются аргументы конечного пути и имени файла. Затем, если заданной , конечной директории не существует, то она будет создана. Далее создается массив строк и ему присваивается значение Directory.GetFiles(sourcePath), добавляющая файлы директории. В цикле foreach, переменной FileName присваивается значение имени файла с помощью метода Path.Getfiles(s). Переменной destFile присваиваем значение директории к файлу. После выполнения всех действий, программа выведет уведомление в textbox1.

Изображение выглядит как текст

Автоматически созданное описание

Удаление каталога:

Изображение выглядит как текст

Автоматически созданное описание

Запуск приложения:

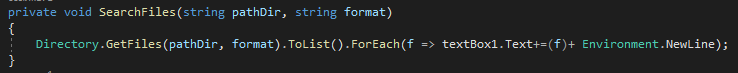
В метод OpenProc, передаем имя файла. В самой процедуре объявляем переменную fileDir типа string и приравниваем ее значение как «соединение 2х строк». С помощью метода Process.Start(fileDir), передаем ему путь к файлу.

Изображение выглядит как текст

Автоматически созданное описание

Поиск файла по расширению:

В функцию SearchFiles(), передаются аргументы string pathDir- указывающую путь к каталогу, string format- отвечающую за формат файла для поиска. В самой процедуре вызывается метод Directory.GetFiles, нужный для поиска в папке файлов, ему передаётся путь и формат файла, с помощью методов .ToList, ForEach и =>, происходит вывод результата в textbox1.



Перемещение файлов:

В функцию MoveFiles(), передаются следующие аргументы: string pathDir-путь к исходной директоии , string name- имя файла с расширением , string pathToCopy- конечный путь перемещения файла. С помощью метода File.Move(), мы передаем ему аргументы пути к исходному файлу и конечный путь копирования.

Изображение выглядит как текст

Автоматически созданное описание

Выбор команды:

В строку string Text передаются команды пользователя. В массив строк string[] words , с помощью метода .Split(‘ ’),команды «режутся» на слова. После, происходит проверка на название команды if(){}else{}, с помощью нулевого элемента массива. Также, происходит обработка исключений с помощью проверки на пустую строку if(words[1].Trim()!=string.Empty), также с помощью try{}catch{} происходит обработка программных исключений.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Вывод об ограничении системы:

