**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2. ИНТЕРПРЕТАТОР КОМАНДНОЙ СТРОКИ ОС MS WINDOWS.**

Часть 2. Язык интерпретатора и командные файлы

**ЦЕЛЬ РАБОТЫ** – знакомство с языком интерпретатора командной строки ОС MS Windows и командными файлами.

**МЕТОДИКА ВЫПОЛНЕНИЯ**

1. Неформально ознакомиться с теоретическими сведениями.
2. Для подготовки текстов командных файлов рекомендуетсся использовать блокнот. При этом следует избегать использования в выводимых на экран результаттах работы командного файла букв русского алфавита.
3. Разработать и выполнить командные файлы, выолняющие следующие функции:
4. Вывод на экран имен всех файловв с указанным расширением, находящихся в каталоге, имя которого задается при запуске командного файла первым параметром. Расширение файлов задается вторым параметром.

**Листинг кода:**

:: z\_04.bat

@echo off

setlocal

rem List file names by extension in a given directory.

rem Usage: z\_04.bat DIR EXT

if "%~1"=="" (echo Usage: %~nx0 DIR EXT & exit /b 1)

if "%~2"=="" (echo Usage: %~nx0 DIR EXT & exit /b 1)

if not exist "%~1" (echo Directory not found: %~1 & exit /b 1)

set "ext=%~2"

if "%ext:~0,1%"=="." set "ext=%ext:~1%"

pushd "%~1" >nul

set "found=0"

for %%F in (\*.%ext%) do (

if exist "%%~fF" (

echo %%~nxF

set "found=1"

)

)

if "%found%"=="0" echo No files with extension .%ext% found

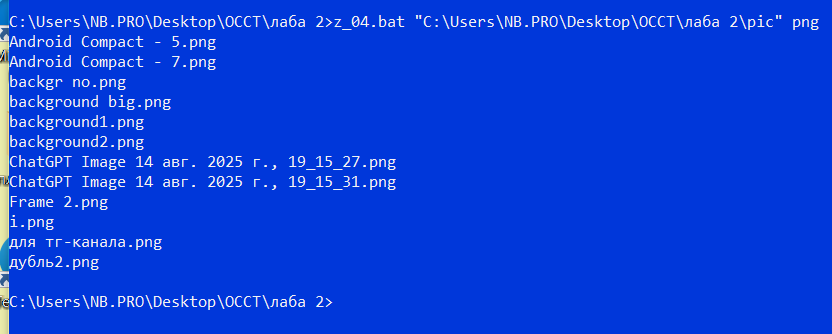
popd >nul

endlocal

В начале стоит **@echo off**, чтобы в консоли не повторялись исполняемые команды, и **setlocal**, чтобы все переменные и настройки жили только внутри этого скрипта.

Программа печатает имена файлов нужного расширения в заданной папке. Две проверки **if "%~1"=="" …** и **if "%~2"=="" …** требуют путь и расширение, следующая **if not exist "%~1" …** подтверждает существование папки. Команда **set "ext=%~2"** берёт расширение, затем **if "%ext:~0,1%"=="."** **set "ext=%ext:~1%"** убирает ведущую точку. **pushd "%~1" >nul** заходит в целевой каталог, **set "found=0"** ставит флаг «не найдено». Цикл **for %%F in (\*.%ext%)** перебирает файлы; внутри цикла **if exist "%%~fF"** страхует, **echo %%~nxF** печатает имя, **set "found=1"** фиксирует, что что-то нашлось. После цикла **if "%found%"=="0" echo No files with extension .%ext% found** сообщает, если совпадений не было. В конце **popd >nul** возвращает исходную папку, endlocal закрывает контекст.

**Готовый результат:**



1. Среди введенных с клавиатуры целых чисел (использовать SET / P) найти наибольшее и наименьшее. Признак конца ввода – знак -.

**Листинг кода:**

:: z\_05.bat

@echo off

setlocal enabledelayedexpansion

rem Read integers until a single '-' line; print MIN and MAX.

echo Enter integers one per line. End with '-' and press Enter.

set "initialized="

:read

set /p v=^>

if "%v%"=="-" goto done

set "v=%v: =%"

if not defined initialized (

set /a min=v

set /a max=v

set "initialized=1"

) else (

set /a t=v

if !t! LSS !min! set /a min=t

if !t! GTR !max! set /a max=t

)

goto read

:done

if not defined initialized (

echo No numbers entered

) else (

echo MIN=!min!

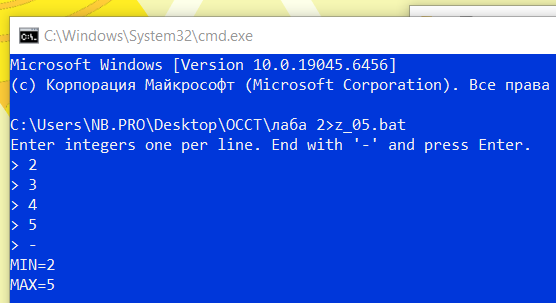
echo MAX=!max!

)

Endlocal

**@echo off** скрывает исполняемые команды, а **setlocal enabledelayedexpansion** включает «отложенное раскрытие» переменных, чтобы внутри цикла корректно обновлять и читать **!min!/!max!**. В программе вводятся числа построчно через команду **set** с ключом **/p**, завершается все строкой «-». Пробелы из ввода удаляются, первое валидное значение инициализирует обе границы, последующие сравниваются через операторы сравнения **LSS/GTR** и при необходимости обновляют минимум и максимум; арифметика выполняется командой **set** с ключом **/a**. По завершении либо печатаются **MIN/MAX**, либо сообщение об отсутствии чисел.

**Готовый результат:**



1. В заданном каталоге и его подкаталогах найти общее количество подкаталогов. На экран вывести только требуемый результат.

Листинг кода:

:: z\_06.bat

@echo off

setlocal

rem Count all subdirectories under a given root (prints number only).

rem Usage: 06\_count\_subdirs.bat [ROOT\_DIR]

set "root=%~1"

if "%root%"=="" set "root=%cd%"

if not exist "%root%" (

echo Directory not found

goto :eof

)

set /a n=0

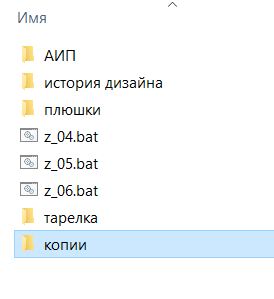
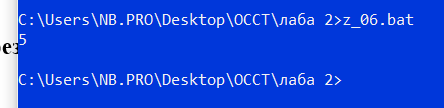
for /r "%root%" /d %%D in (\*) do set /a n+=1

echo %n%

endlocal

В начале стоит **@echo off**, чтобы в консоли не повторялись исполняемые команды, и **setlocal**, чтобы все переменные и настройки жили только внутри этого скрипта. Программа берёт корень из параметра или **%cd%**, проверяет путь, запускает счётчик и одной командой цикла **for** с ключами **/r /d** рекурсивно проходит подкаталоги, увеличивая счётчик на каждом найденном каталоге. После проверки выводится только число, без лишнего текста. В программе учтены все уровни вложенности, сам корень при этом не добавляется.

**Готовый результат:**

1. В каталогах, имена которых заданы первым и вторым параметрами командного файла, найти и вывести на экран имена файлов (расширения могут быть любые), присутствующи как в первом, так и во втором каталоге. Следует использовать только один оператор FOR.

Листинг кода:

:: z\_07.bat

@echo off

setlocal

rem Print file names that exist in both directories (top-level only). Uses exactly one FOR.

rem Usage: z\_07.bat DIR1 DIR2

if "%~1"=="" (

echo Usage: %~nx0 DIR1 DIR2

goto :eof

)

if "%~2"=="" (

echo Usage: %~nx0 DIR1 DIR2

goto :eof

)

if not exist "%~1" (

echo Directory not found: %~1

goto :eof

)

if not exist "%~2" (

echo Directory not found: %~2

goto :eof

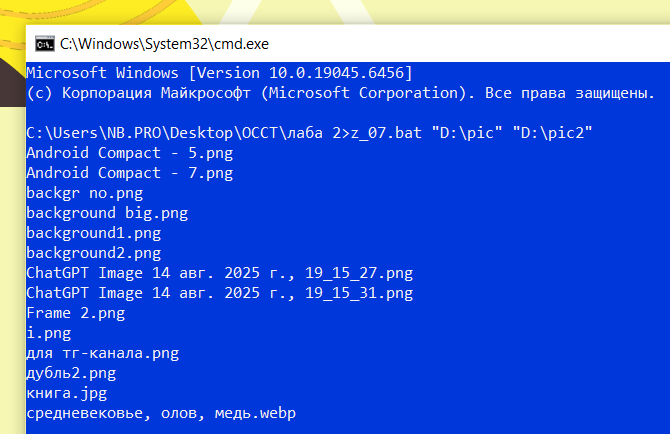
)

for /f "delims=" %%F in ('dir /b /a:-d "%~1"') do if exist "%~2\%%~nxF" echo %%~nxF

endlocal

Программа проверяет оба пути и единственной конструкцией **for /f "delims=" %%F in ('dir /b /a:-d "%~1"')** построчно читает одноуровневый список файлов первого каталога (формат /b, без подкаталогов благодаря /a:-d), т.е. оставляет только файлы. **Delims** отключает разбиение по пробелам, чтобы имена с пробелами попадали целиком. Для каждого имени выполняется проверка **if exist "%~2\%%~nxF"** и при совпадении выводится имя файла. Сравнение идёт только по совпадению имён в верхнем уровне каталогов, содержимое и размеры не проверяются.

**Готовый результат:**



1. Вычисление и вывод на экран значения факторала целого числа, задаваемого при запуске КФ. Предусмотреть проверку заданного значения и при задании отрицательного значения или значения, превышающего максимально возможную величину, выводить соответствующие сообщения. Для проверки правильности вычислений использовать калькулятор.

**Листинг кода:**

:: z\_08.bat

@echo off

setlocal enabledelayedexpansion

rem Compute factorial of N with 32-bit bound checking (fits up to 12!).

rem Usage: z\_08.bat N

if "%~1"=="" (

echo Usage: %~nx0 N

goto :eof

)

set /a n=%~1

if %n% LSS 0 (

echo Negative values are not allowed

goto :eof

)

if %n% GTR 12 (

echo Value too large for 32-bit integer. Max N is 12.

goto :eof

)

set /a f=1

for /l %%i in (2,1,%n%) do set /a f\*=%%i

set "res=!f!"

setlocal DisableDelayedExpansion

echo %n%^! = %res%

endlocal

В начале стоит **@echo off**, чтобы в консоли не повторялись исполняемые команды, и **setlocal**, чтобы все переменные и настройки жили только внутри этого скрипта. Программа читает **N** арифметически (команда set /a n=%~1), отсекает пустой параметр, отрицательные значения и слишком большие (>12) из-за 32-битной арифметики команды **set** с ключом **/a**, инициализирует **f=1**, затем циклом **for /l** последовательно умножает на все числа **от 2 до N**; выводит строку вида **N! = результат**, где актуальное значение берётся через **!f!** при включённой команде **enabledelayedexpansion**.

**Готовый результат:**

