Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

|  |  |
| --- | --- |
| Институт | математики и компьютерных наук |
| Кафедра | компьютерной безопасности |

ОТЧЕТ

о Лабораторная работа №1

по дисциплине «Интерпретатор командной строки ОС MS Windows»

Выполнил:

Пронин Владимир Иванович,

студент 2 курса

группы КМБ-с-о-23-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Проверено с оценкой:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

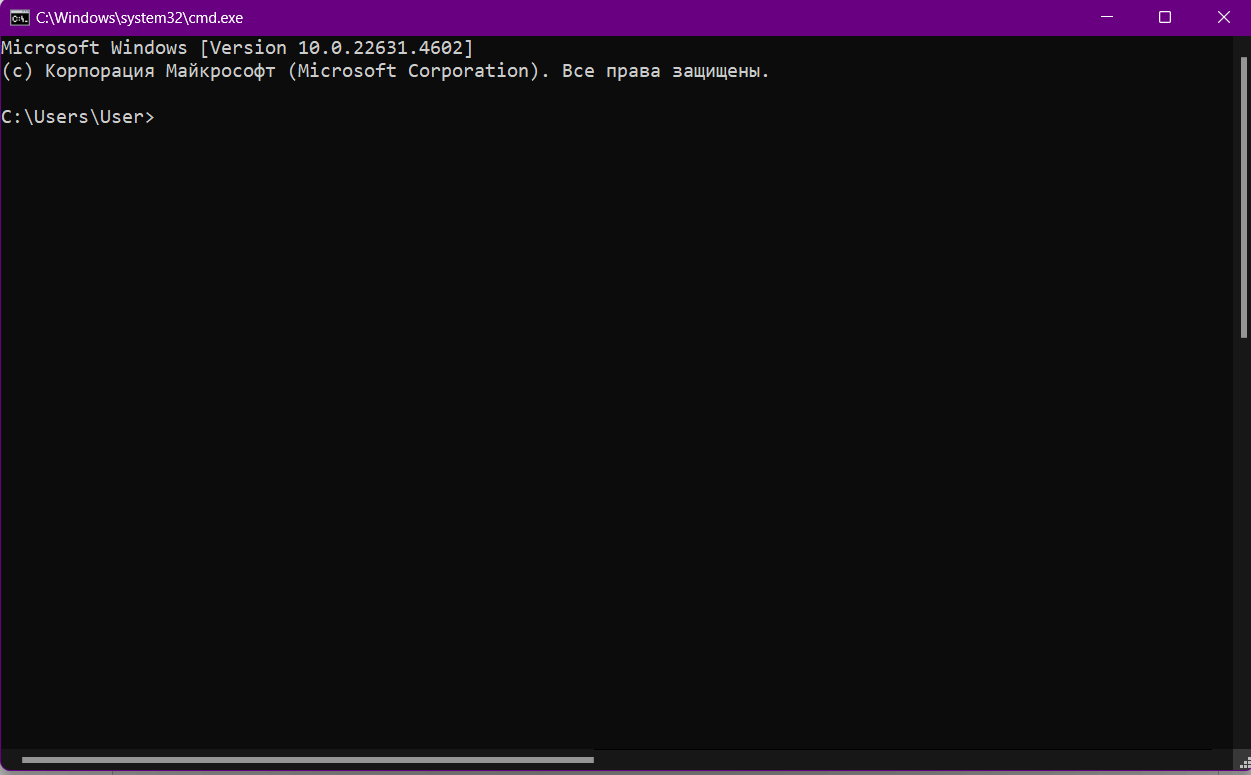
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

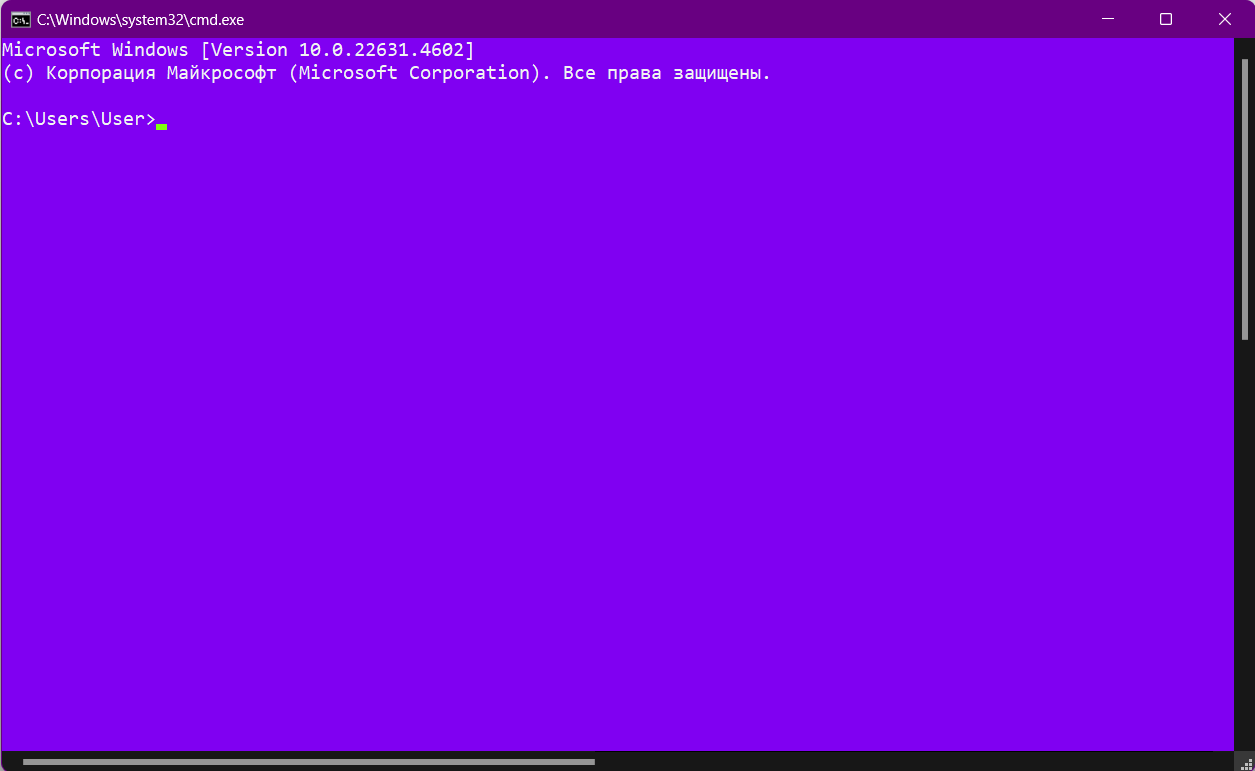
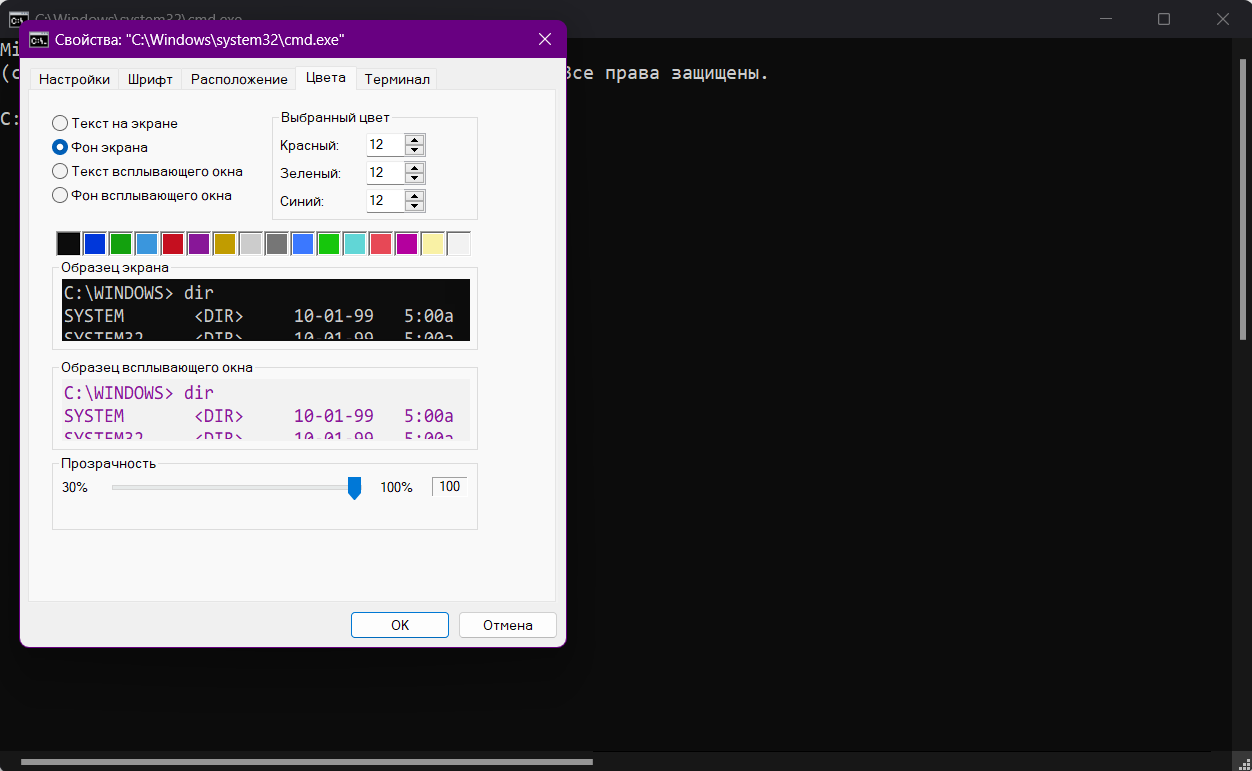
«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

Ставрополь, 2023

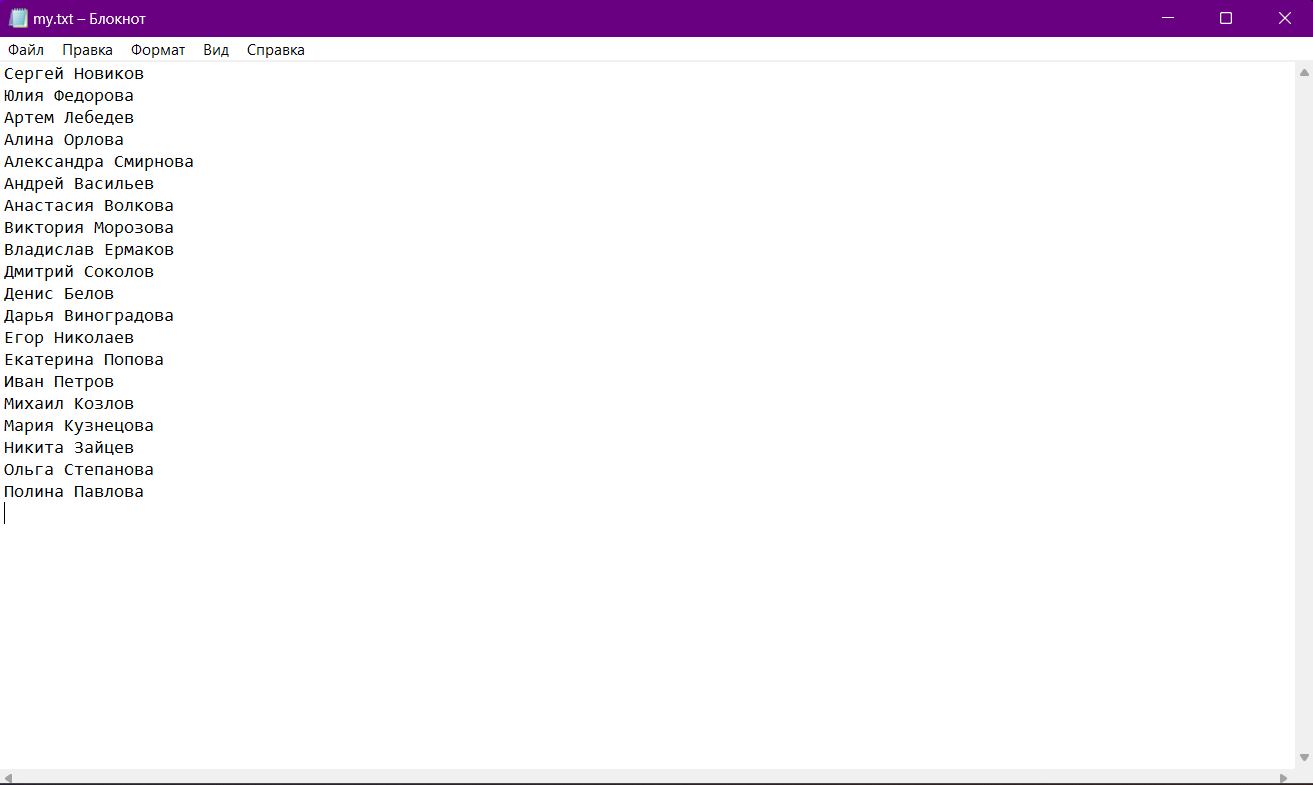
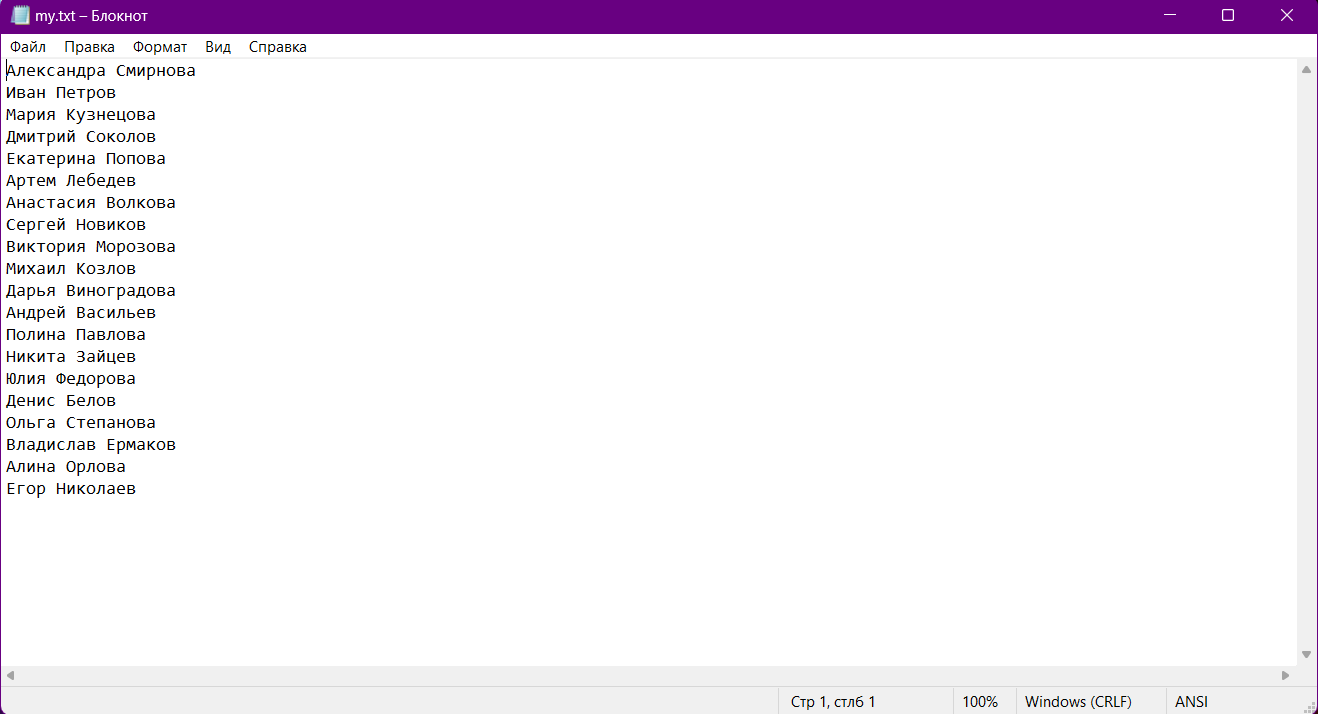
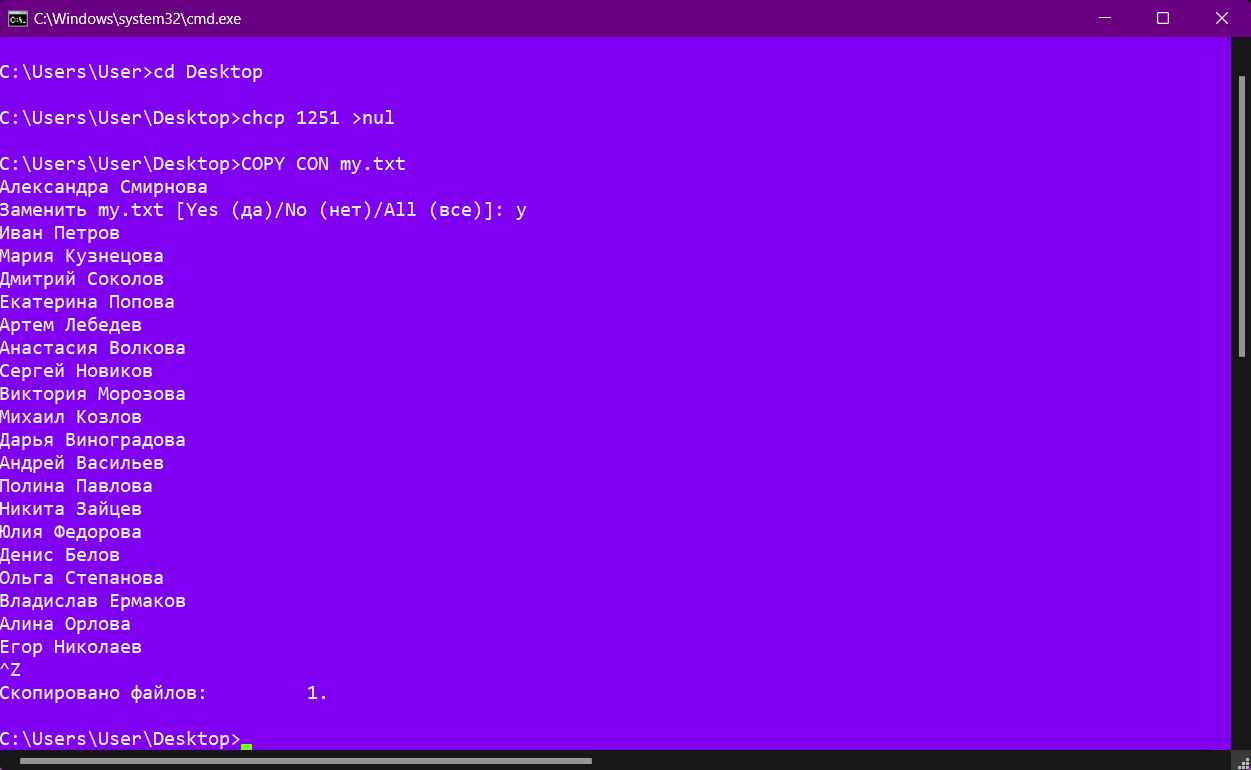
2) Запустить интерпретатор командной строки



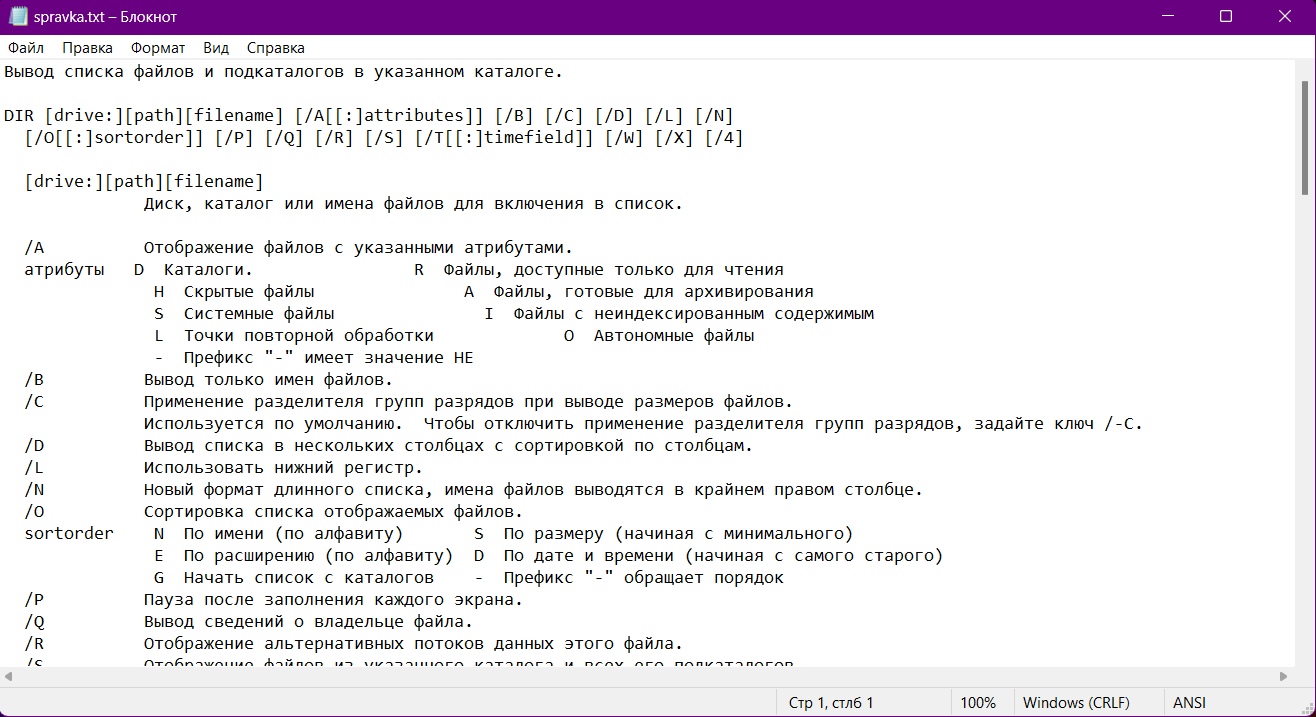
3) Увеличить размер окна интерпретатора и задать цвет фона и цвет шрифта (рекомендуется синий фон и белый шрифт).



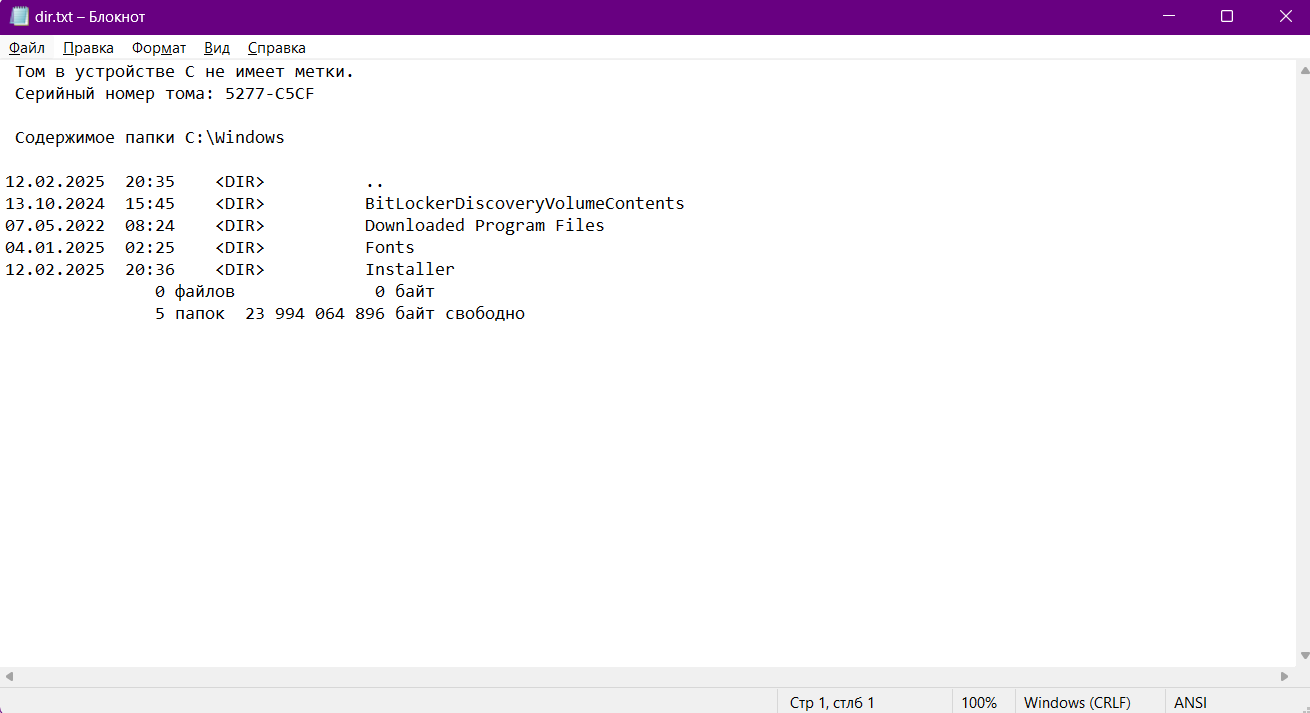
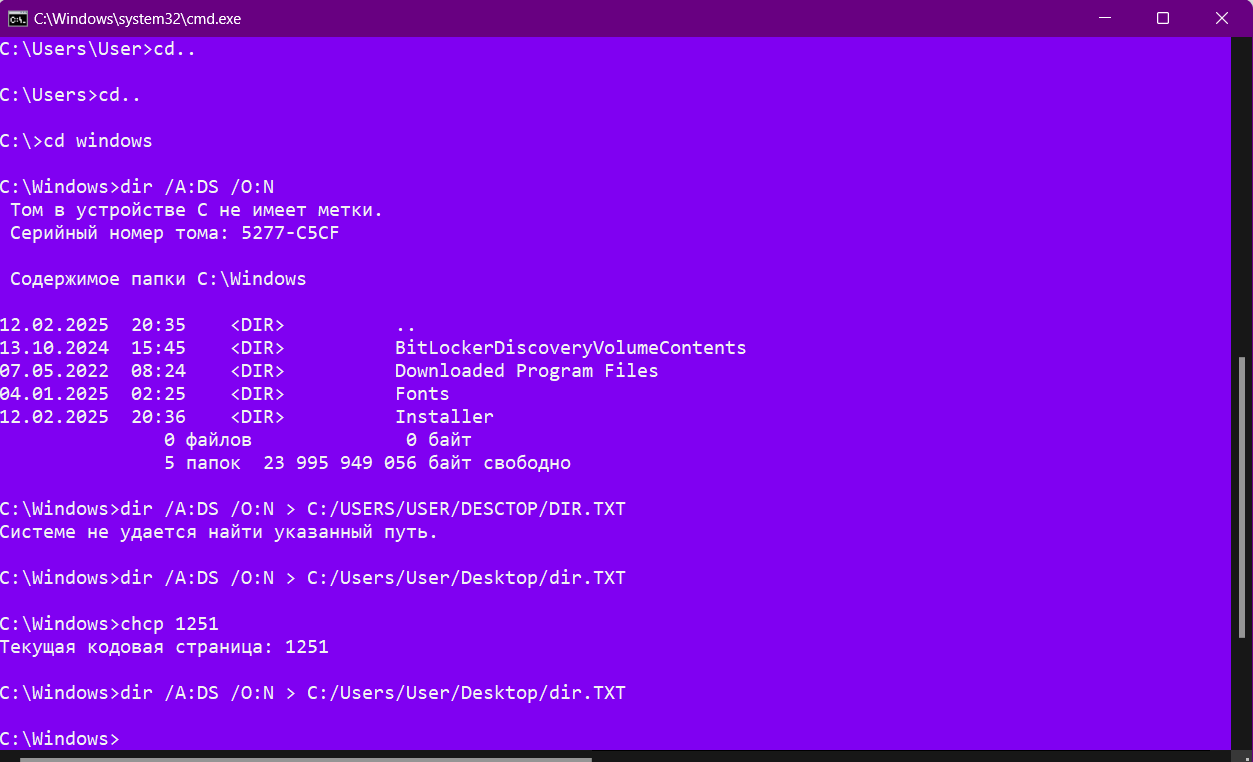
4) Создать список фамилий студентов группы, используя пример 1. Отсортировать список в алфавитном порядке и сохранить его в новом файле



5) Создать текстовый файл, содержащий справочные сведения по командам DIR, COPY и XCOPY.

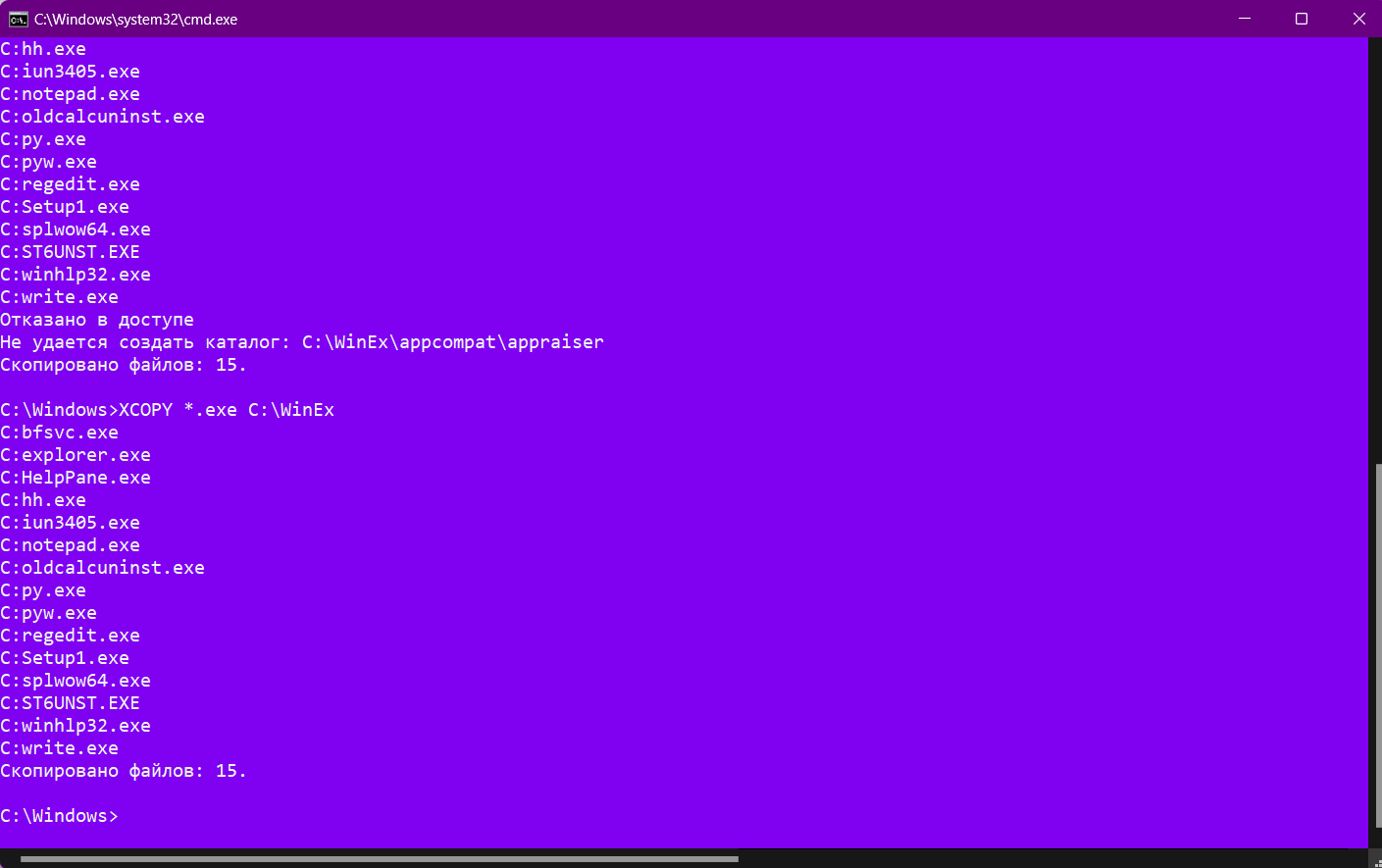


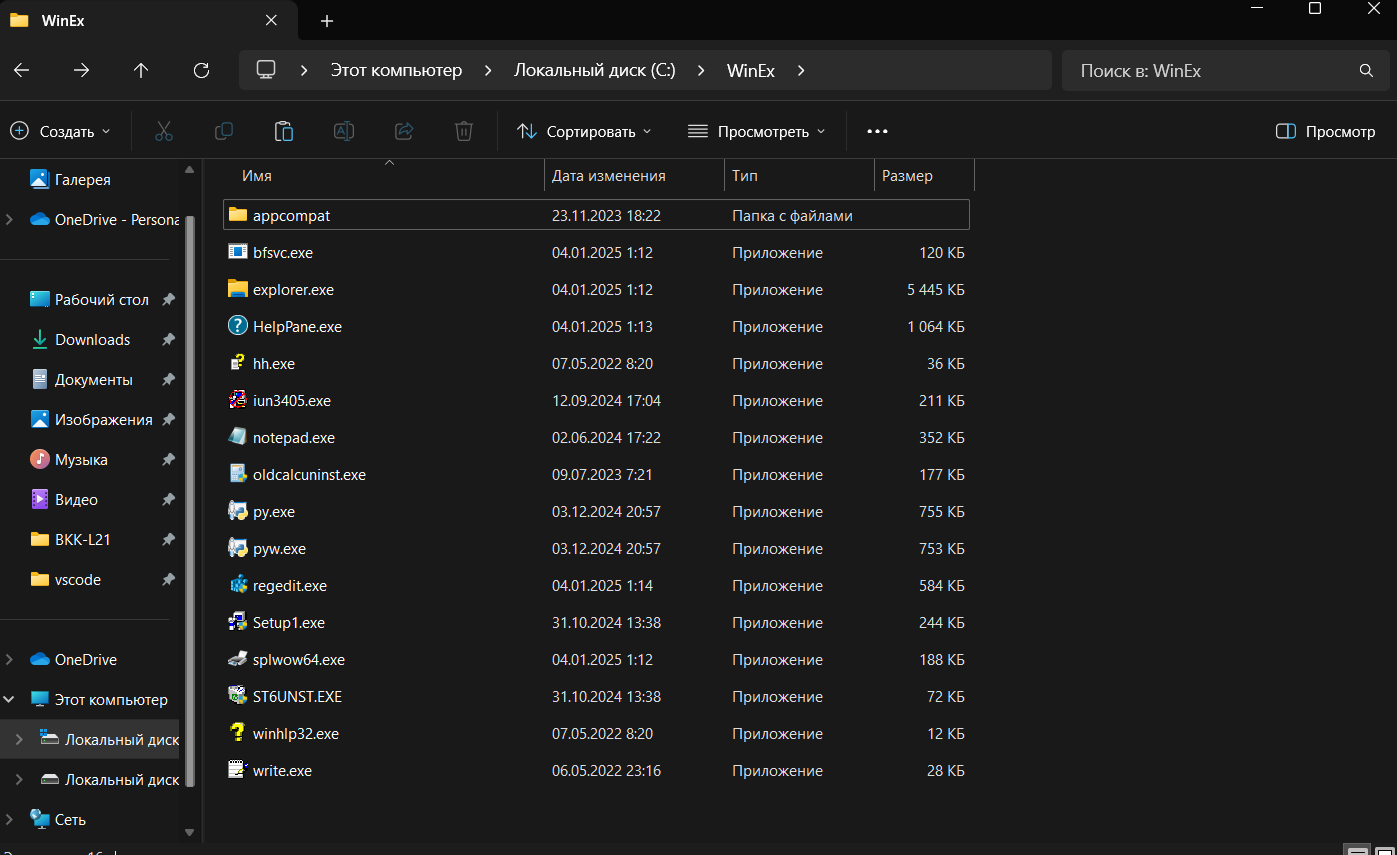
6) Вывести содержимое указанного в табл.3 каталога по указанному формату на экран и в файл.

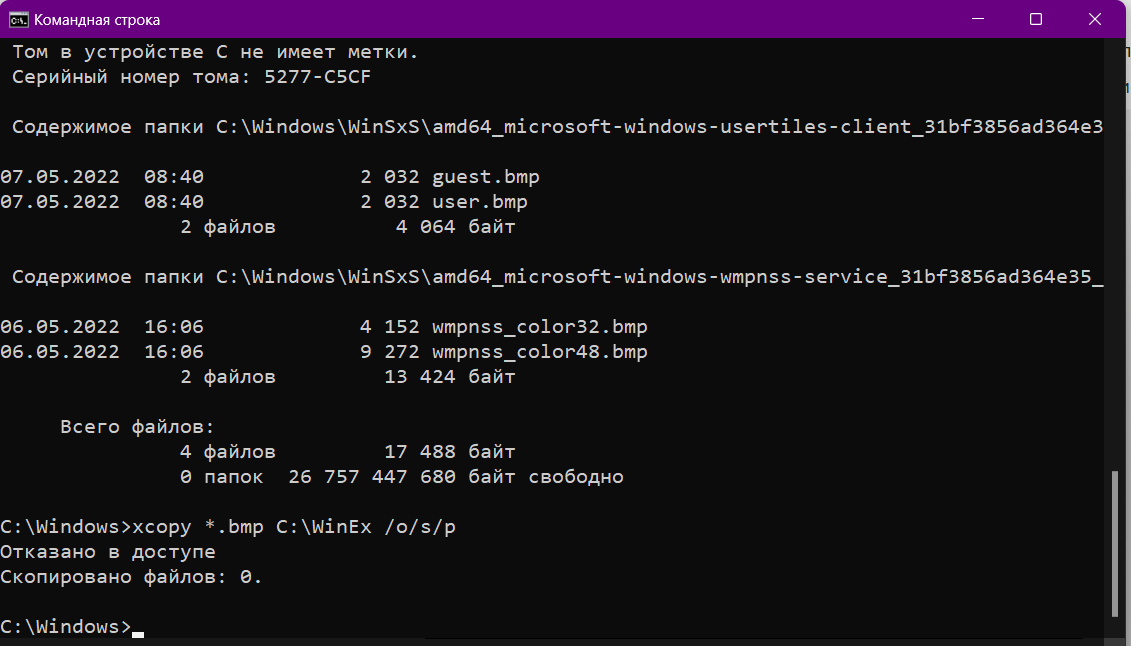


|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Номера вариантов | Имя каталога | Что выводить | Сортировать по | Атрибуты файлов и каталогов |
| 7 | %Windows% | Только подкаталоги | Именам | Системный |

8)Скопировать все имеющиеся в каталоге Windows растровые графические файлы в каталог WinGrafika на диске С:. Если диск С: недоступен, использовать любой другой доступный диск.







Вопросы? :|

**1. Достоинства и недостатки интерфейса командной строки (CLI)**

**Достоинства:**

* **Мощность и гибкость:** CLI предоставляет прямой доступ к функциональности операционной системы и позволяет выполнять сложные операции, часто недоступные через графический интерфейс (GUI).
* **Автоматизация:** CLI отлично подходит для автоматизации задач с помощью скриптов (например, bat-файлов в Windows, bash-скриптов в Linux). Можно создавать последовательности команд для выполнения рутинных операций.
* **Эффективность:** Для многих задач CLI может быть быстрее и эффективнее, чем GUI, особенно для опытных пользователей. Не нужно перемещать мышь, кликать по меню и диалоговым окнам.
* **Удаленное управление:** CLI является основным способом управления серверами и другими удаленными системами, особенно в тех случаях, когда графический интерфейс недоступен или нецелесообразен.
* **Низкие требования к ресурсам:** CLI требует значительно меньше ресурсов (процессор, память) по сравнению с GUI. Это особенно важно для старых или маломощных компьютеров.
* **Точность:** CLI позволяет точно указать параметры и аргументы для каждой команды, что исключает двусмысленность, которая может возникнуть в GUI.

**Недостатки:**

* **Сложность обнаружения:** Трудно узнать, какие команды доступны и как их использовать, если нет справочной системы или документации. В GUI обычно можно обнаружить функциональность, просто просматривая меню.
* **Ошибки:** Опечатки в командах или неправильные параметры могут привести к непредсказуемым результатам или ошибкам.
* **Сложность визуализации:** CLI не подходит для задач, требующих визуального представления данных или интерактивного взаимодействия с графическими элементами.
* **Недружелюбный интерфейс:** CLI может показаться менее интуитивным и дружелюбным по сравнению с GUI.
* **Зависимость от команд:** Пользователь должен помнить и знать команды, нет возможности увидеть варианты работы в графическом виде.

**2. Инструменты командной строки для автоматизации работы в ОС Microsoft Windows**

В Windows есть несколько инструментов командной строки, которые можно использовать для автоматизации работы:

* **cmd.exe (Командная строка):**
  + Базовый интерпретатор команд, доступный во всех версиях Windows.
  + Использует bat-файлы для создания скриптов.
  + Поддерживает основные команды для управления файлами, каталогами, процессами и системой.
  + Ограниченные возможности по сравнению с PowerShell.
* **PowerShell:**
  + Более мощный и современный интерпретатор команд.
  + Использует язык PowerShell, основанный на .NET.
  + Предоставляет доступ к огромному количеству командлетов (cmdlets) для управления практически всеми аспектами Windows.
  + Поддерживает продвинутые функции, такие как конвейеры (pipelines), объекты и скрипты.
  + Рекомендуется для большинства задач автоматизации в Windows.
* **Windows Script Host (WSH):**
  + Среда выполнения скриптов, которая позволяет запускать скрипты, написанные на различных языках (например, VBScript, JScript).
  + Может использоваться для создания скриптов автоматизации, взаимодействующих с Windows API и COM-объектами.
  + Сейчас считается устаревшим, но все еще может быть полезен для некоторых задач.

**3. Настраиваемые свойства интерпретатора (cmd.exe)**

Интерпретатор командной строки cmd.exe имеет несколько настраиваемых свойств, которые можно изменить для улучшения удобства использования:

* **Размер и положение окна:** Можно настроить размер и положение окна cmd.exe.
* **Шрифт:** Можно выбрать шрифт, используемый в окне cmd.exe. Важно выбрать шрифт, который поддерживает нужную кодировку (например, кириллицу).
* **Цвета:** Можно изменить цвета текста, фона и других элементов окна cmd.exe.
* **Размер буфера экрана:** Определяет количество строк текста, которое хранится в буфере окна cmd.exe. Можно увеличить этот размер, чтобы можно было прокручивать больше текста.
* **История команд:** cmd.exe сохраняет историю последних введенных команд. Можно настроить количество команд, которые сохраняются в истории.
* **Сочетания клавиш:** Можно настроить сочетания клавиш для выполнения различных действий (например, копирования, вставки, очистки экрана).
* **Автозавершение имен файлов и каталогов:** cmd.exe может автоматически завершать имена файлов и каталогов при нажатии клавиши Tab.

**Как настроить свойства интерпретатора:**

1. Щелкните правой кнопкой мыши по заголовку окна cmd.exe.
2. Выберите “Свойства”.
3. В диалоговом окне “Свойства” можно изменить различные настройки, такие как шрифт, цвета, размер окна, и т.д.

**4. Различие между внутренними и внешними командами. Примеры**

Команды в cmd.exe (и в других командных интерпретаторах) делятся на две категории: внутренние и внешние.

* **Внутренние команды:**
  + Встроены непосредственно в интерпретатор командной строки (cmd.exe).
  + Загружаются в память вместе с интерпретатором.
  + Выполняются быстро, так как не требуют загрузки отдельных файлов с диска.
  + Всегда доступны, независимо от текущего каталога.
  + Примеры: cd, dir, echo, md (или mkdir), rd (или rmdir), copy, del, ren, type, cls, exit, set, title, ver, date, time, prompt.
* **Внешние команды:**
  + Хранятся в отдельных исполняемых файлах (обычно с расширением .exe, .com, .bat, .cmd).
  + Загружаются в память и выполняются только при вызове.
  + Для выполнения внешней команды необходимо, чтобы файл команды находился в текущем каталоге или в одном из каталогов, указанных в переменной среды PATH.
  + Примеры: xcopy, attrib, diskpart, find, findstr, ipconfig, ping, sort, tasklist, wmic, format, chkdsk.

**5. Структура команды интерпретатора (cmd.exe)**

Команда в cmd.exe обычно имеет следующую структуру:

команда [параметры] [аргументы]

* **команда:** Имя команды, которую нужно выполнить (например, dir, copy, echo).
* **параметры (опции, ключи):** Модификаторы команды, которые изменяют ее поведение. Обычно начинаются с символа / (например, /w, /p, /s). Параметры могут быть обязательными или необязательными.
* **аргументы:** Данные, которые передаются команде. Они могут быть именами файлов, каталогов, текста, чисел и т.д. Аргументы могут быть обязательными или необязательными.
* **Разделители:** Пробелы обычно используются в качестве разделителей между командой, параметрами и аргументами.
* **Перенаправление ввода/вывода:** Используются символы >, <, >>, | для перенаправления ввода и вывода команды.

**Примеры:**

* dir /w C:\Windows:
  + dir: Команда (отобразить содержимое каталога).
  + /w: Параметр (отобразить в широком формате).
  + C:\Windows: Аргумент (каталог, содержимое которого нужно отобразить).
* copy file1.txt file2.txt:
  + copy: Команда (копировать файл).
  + file1.txt: Аргумент (исходный файл).
  + file2.txt: Аргумент (целевой файл).
* echo "Hello, world!" > output.txt:
  + echo: Команда (отобразить текст).
  + "Hello, world!": Аргумент (текст для отображения).
  + >: Перенаправление вывода (записать текст в файл output.txt).
* type file.txt | find "keyword":
  + type: Команда (отобразить содержимое файла).
  + file.txt: Аргумент (файл для отображения).
  + |: Конвейер (передать вывод type на вход find).
  + find: Команда (поиск текста в потоке ввода).
  + "keyword": Аргумент (текст для поиска).

**6. Справка о команде (Узнать, как использовать команду):**

* **cmd:** help <команда> или <команда> /? (простая справка).
  + Пример: help dir или dir /? - покажет справку по команде dir.

**7. Групповые символы (шаблоны) (Выбрать несколько файлов/каталогов по маске):**

* **Общие:** \* (любое количество символов), ? (один любой символ).
  + Пример: dir \*.txt - покажет все файлы с расширением .txt в текущем каталоге (в cmd.exe).

**8. Перенаправление ввода/вывода и конвейер (Перенаправить результаты команд или соединить их):**

* **Общие:** > (перезаписать файл), >> (добавить в файл), < (читать из файла), | (передать вывод одной команды на вход другой).
  + Пример: dir > filelist.txt - запишет список файлов в текущем каталоге в файл filelist.txt (в cmd.exe).

**9. Условное выполнение и группировка (Выполнить команды, если условие выполнено/не выполнено):**

* **cmd:** && (И: выполнить, если предыдущая команда успешна), || (ИЛИ: выполнить, если предыдущая команда неудачна), () (группировать команды), IF (полноценный условный оператор).
  + Пример: mkdir NewDir && cd NewDir - создаст каталог NewDir и перейдет в него только если создание прошло успешно (в cmd.exe).

**10. Символы &, &&, ||, () (Управление выполнением команд):**

* **cmd:**
  + & (редко) - выполнить команды последовательно (обычно не нужен). Пример: dir & echo "Done"
  + && (И) - условное выполнение: выполнить следующую команду только если предыдущая успешна. Пример: mkdir NewDir && cd NewDir
  + || (ИЛИ) - условное выполнение: выполнить следующую команду только если предыдущая *не* успешна. Пример: del file.txt || echo "File not found"
  + () - сгруппировать команды для перенаправления вывода (например, в файл). Пример: (dir && echo "Success") > output.txt

**11. Команды для работы с файловой системой – названия и возможности (cmd.exe и PowerShell)**

**cmd.exe:**

* **DIR (Directory):** Выводит список файлов и подкаталогов в указанном каталоге. Поддерживает параметры для сортировки, фильтрации и отображения информации.
* **CD (Change Directory):** Изменяет текущий каталог.
* **MD или MKDIR (Make Directory):** Создает новый каталог.
* **RD или RMDIR (Remove Directory):** Удаляет каталог (только пустой, если не указан параметр /S).
* **COPY:** Копирует один или несколько файлов в другое место. Ограниченные возможности по сравнению с XCOPY.
* **XCOPY:** Расширенная версия команды COPY, которая позволяет копировать каталоги, подкаталоги, атрибуты файлов и многое другое.
* **DEL (Delete):** Удаляет один или несколько файлов.
* **REN (Rename):** Переименовывает файл или каталог.
* **TYPE:** Выводит содержимое текстового файла на экран.
* **ATTRIB (Attribute):** Изменяет или отображает атрибуты файлов (например, “ReadOnly”, “Hidden”, “System”, “Archive”).
* **MOVE:** Перемещает файлы или каталоги из одного места в другое. Может также переименовывать файлы и каталоги.

**PowerShell:**

* **Get-ChildItem:** (Аналог DIR) Выводит список файлов и подкаталогов в указанном месте. Поддерживает мощные возможности для фильтрации, сортировки и отображения информации.
* **Set-Location:** (Аналог CD) Изменяет текущее местоположение (каталог).
* **New-Item:** Создает новый элемент (файл, каталог, ссылка и т.д.). Для создания каталога используйте -ItemType Directory.
* **Remove-Item:** (Аналог DEL, RD) Удаляет один или несколько элементов (файлов, каталогов и т.д.). Поддерживает параметр -Recurse для удаления непустых каталогов.
* **Copy-Item:** (Аналог COPY, XCOPY) Копирует один или несколько элементов в другое место. Поддерживает параметры для копирования каталогов, атрибутов и т.д.
* **Rename-Item:** (Аналог REN) Переименовывает элемент.
* **Get-Content:** (Аналог TYPE) Получает содержимое файла.
* **Set-Content:** (Записывает содержимое в файл, перезаписывая его)
* **Add-Content:** (Добавляет содержимое в конец файла)
* **Get-ItemProperty:** (Получает свойства элемента, в том числе атрибуты).
* **Set-ItemProperty:** (Изменяет свойства элемента, в том числе атрибуты).
* **Move-Item:** (Аналог MOVE) Перемещает элементы.

**Основные различия:**

* PowerShell использует командлеты (cmdlets), а cmd.exe - команды.
* PowerShell работает с объектами, а cmd.exe - в основном с текстом.
* PowerShell предоставляет более мощные и гибкие возможности для фильтрации, сортировки и обработки данных.

**12. Достоинства и недостатки команд COPY и XCOPY (cmd.exe)**

Обе команды используются для копирования файлов, но XCOPY предлагает расширенные возможности.

**COPY:**

* **Достоинства:**
  + Простая и понятная команда.
  + Доступна во всех версиях Windows.
  + Быстрая для копирования отдельных файлов.
* **Недостатки:**
  + Ограниченные возможности. Не может копировать каталоги или подкаталоги.
  + Не сохраняет атрибуты файлов (например, дату и время создания).
  + Не может копировать файлы по сети в режиме перезапуска.

**XCOPY:**

* **Достоинства:**
  + Расширенные возможности: копирование каталогов и подкаталогов, сохранение атрибутов, копирование файлов по сети в режиме перезапуска.
  + Более гибкая настройка с помощью параметров (например, /S, /E, /H, /K, /Y, /Z).
* **Недостатки:**
  + Более сложный синтаксис, чем у COPY.
  + Может быть медленнее, чем COPY, при копировании отдельных файлов.
  + Менее интуитивна, чем COPY.

**Когда использовать:**

* Используйте COPY для быстрого копирования отдельных файлов, когда не требуется сохранение атрибутов или копирование каталогов.
* Используйте XCOPY для копирования каталогов, подкаталогов и файлов с сохранением атрибутов, а также для копирования файлов по сети в режиме перезапуска. XCOPY - предпочтительный вариант для большинства задач копирования.

**PowerShell:** Copy-Item (аналог XCOPY, но с объектами и более мощными возможностями).

**13. Назначение команды ECHO и примеры ее использования (cmd.exe)**

**Назначение:**

Команда ECHO используется для отображения текста на экране или для управления состоянием вывода команд.

**Примеры:**

* **echo Hello, world!:** Отображает текст “Hello, world!” на экране.
* **echo. или echo:** Выводит пустую строку. Используется для добавления пустых строк в вывод или в файлы.
* **echo > file.txt:** Создает пустой файл с именем file.txt. Если файл уже существует, он будет перезаписан.
* **echo Some text > file.txt:** Создает файл file.txt с содержимым “Some text”. Если файл уже существует, он будет перезаписан.
* **echo Some text >> file.txt:** Добавляет текст “Some text” в конец файла file.txt. Если файл не существует, он будет создан.
* **echo %variable%:** Отображает значение переменной среды. Например, echo %PATH% отобразит значение переменной PATH.
* **echo off:** Отключает отображение команд в командной строке (используется в bat-файлах).
* **echo on:** Включает отображение команд в командной строке (используется в bat-файлах).
* **@echo off:** Отключает отображение команды @echo off в командной строке (используется в начале bat-файлов для предотвращения вывода самой команды echo off).

**PowerShell:** Используйте командлет Write-Host (для вывода на экран) или Write-Output (для вывода в конвейер или файл).

**14. Команда DIR и ее возможности (cmd.exe)**

**Назначение:**

Команда DIR используется для вывода списка файлов и подкаталогов в указанном каталоге.

**Возможности (параметры):**

* **DIR:** Выводит список файлов и каталогов в текущем каталоге.
* **DIR <путь>:** Выводит список файлов и каталогов в указанном каталоге.
  + DIR C:\Windows: Выводит список файлов и каталогов в каталоге C:\Windows.
* **DIR /P:** Выводит список файлов и каталогов постранично.
* **DIR /W:** Выводит список файлов и каталогов в широком формате (в несколько колонок).
* **DIR /AD:** Выводит только подкаталоги.
* **DIR /AH:** Выводит только скрытые файлы и каталоги.
* **DIR /AS:** Выводит только системные файлы и каталоги.
* **DIR /A:** Выводит файлы и каталоги с указанными атрибутами (можно комбинировать атрибуты, например, /AHS для скрытых и системных).
* **DIR /O:<сортировка>:** Сортирует список файлов и каталогов по указанному критерию.
  + /O:N: Сортировка по имени.
  + /O:E: Сортировка по расширению.
  + /O:D: Сортировка по дате.
  + /O:S: Сортировка по размеру.
  + Можно использовать - перед критерием для обратной сортировки (например, /O:-N для сортировки по имени в обратном порядке).
* **DIR /S:** Выводит список файлов и подкаталогов во всех подкаталогах (рекурсивно).
* **DIR /B:** Выводит только имена файлов и каталогов (без дополнительной информации).
* **DIR /L:** Использует нижний регистр для имен файлов и каталогов.
* **DIR /Q:** Отображает владельца файла.
* **DIR /T:<время>:** Определяет, какое время отображать и использовать для сортировки.
  + /T:C: Время создания.
  + /T:A: Время последнего доступа.
  + /T:W: Время последнего изменения (по умолчанию).
* **DIR <имя\_файла>:** Выводит информацию только о файле с указанным именем (или файлах, соответствующих шаблону с групповыми символами, например DIR \*.txt).

**Примеры:**

* DIR C:\Windows /AD: Вывести список только подкаталогов в C:\Windows.
* DIR \*.txt /O:N: Вывести список файлов с расширением .txt, отсортированных по имени.
* DIR /S /B > filelist.txt: Создать текстовый файл filelist.txt со списком всех файлов и каталогов в текущем каталоге и всех его подкаталогах (только имена, без дополнительной информации).

**PowerShell:** Используйте командлет Get-ChildItem для аналогичных задач.

**15. В какой кодировке интерпретатор выводит информацию и как получить читаемую твердую копию?**

**Кодировка вывода интерпретатора:**

* **cmd.exe:** По умолчанию использует кодировку, установленную для системы (обычно OEM-кодировка, например, CP866 для русского языка). Это историческая кодировка для DOS, у которой есть ограничения по символам и их отображению.
* **PowerShell:** По умолчанию использует кодировку UTF-16LE (Unicode), но может использовать и другие кодировки в зависимости от настроек системы и командлетов.

**Проблемы с кодировкой:**

Если кодировка консоли не соответствует кодировке символов, которые нужно отобразить, могут возникнуть проблемы с отображением (неправильные символы, квадратики и т.д.). Это особенно актуально для cmd.exe.

**Как получить читаемую твердую копию (распечатку) с русским текстом:**

1. **Измените кодировку консоли (cmd.exe):**
   * Используйте команду chcp 1251 для установки кодировки Windows-1251 (Cyrillic). Это наиболее распространенная кодировка для русского языка в Windows.
   * Измените шрифт консоли на шрифт, поддерживающий кириллицу (например, Lucida Console, Consolas, Courier New). Щелкните правой кнопкой мыши по заголовку окна консоли, выберите “Свойства”, перейдите на вкладку “Шрифт” и выберите подходящий шрифт.
2. **Сохраните вывод в файл:**
   * Перенаправьте вывод команды в файл: dir > filelist.txt или type myfile.txt > output.txt.
   * В PowerShell используйте Out-File -Encoding UTF8 или Export-Csv -Encoding UTF8, чтобы указать кодировку файла. Например: Get-ChildItem | Out-File filelist.txt -Encoding UTF8
   * **Важно:** Выберите кодировку UTF-8, если хотите, чтобы файл был совместим с другими системами и приложениями.
3. **Откройте файл в текстовом редакторе и распечатайте:**
   * Откройте файл в текстовом редакторе, поддерживающем выбранную кодировку (например, Notepad++, Visual Studio Code).
   * Убедитесь, что текстовый редактор правильно распознает кодировку файла. Если нет, выберите правильную кодировку вручную (например, “UTF-8” или “Windows-1251”).
   * Распечатайте файл из текстового редактора.
4. **PowerShell (печать напрямую):**
   * В PowerShell можно использовать командлет Out-Printer, чтобы напрямую отправить вывод на принтер. Однако, могут возникнуть проблемы с кодировкой, поэтому рекомендуется сначала сохранить вывод в файл с кодировкой UTF-8 и затем распечатать файл.
5. Get-ChildItem | Out-File -FilePath temp.txt -Encoding UTF8
6. Get-Content temp.txt | Out-Printer

Remove-Item temp.txt

powershell