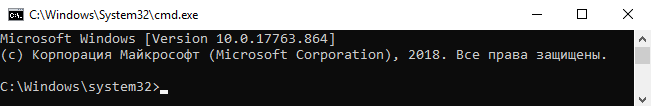
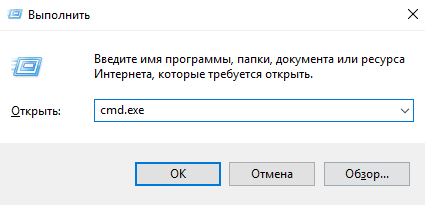
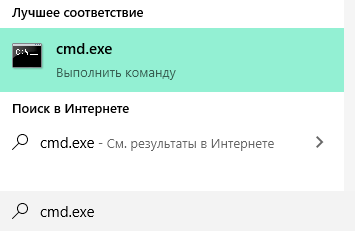
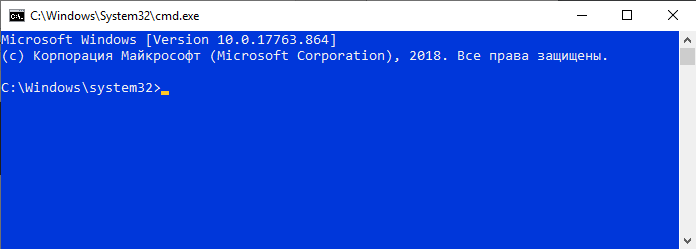
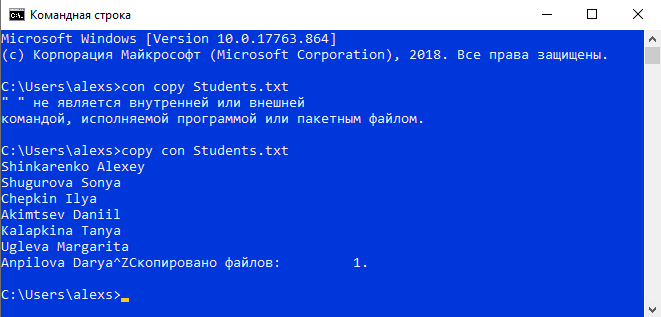
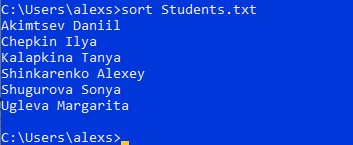
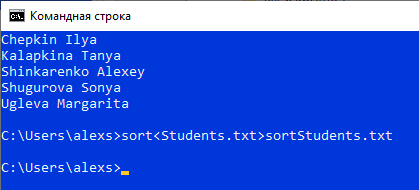
1. Ознакомился с теоритеческими сведениями
2. Запустил cmd.exe через win+r (горячая клавиша для открытие окна «Выполнить») и еще смог запустить через поиск, введя там cmd.exe



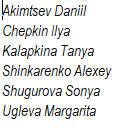
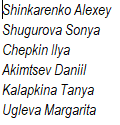
1. Увеличил размер окна. Цвет фона на синий, а цвет шрифта на белый через свойства.
2. Создал список студентов прервали Ctrl+z нажали ctrl+c

Отсортировал список sort НазваниеБлокнота.txt

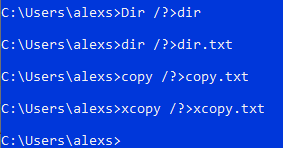


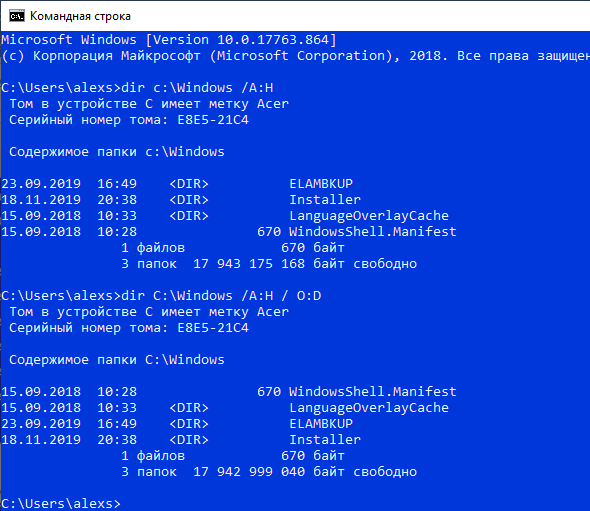
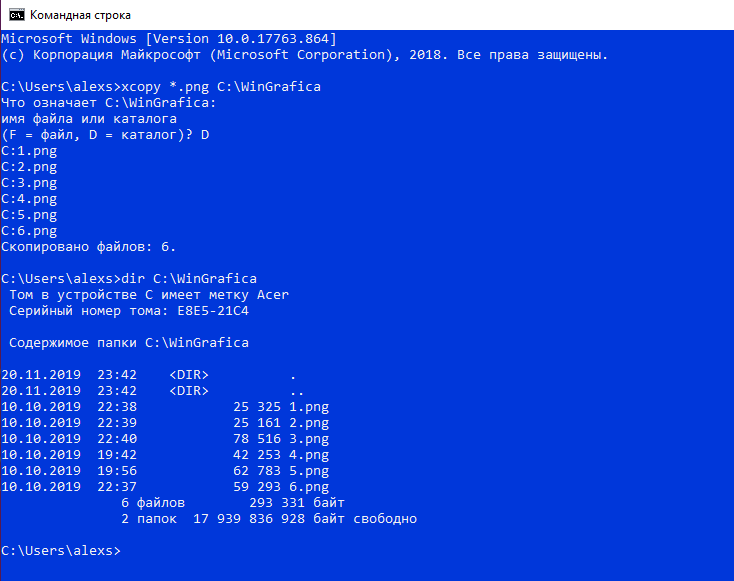
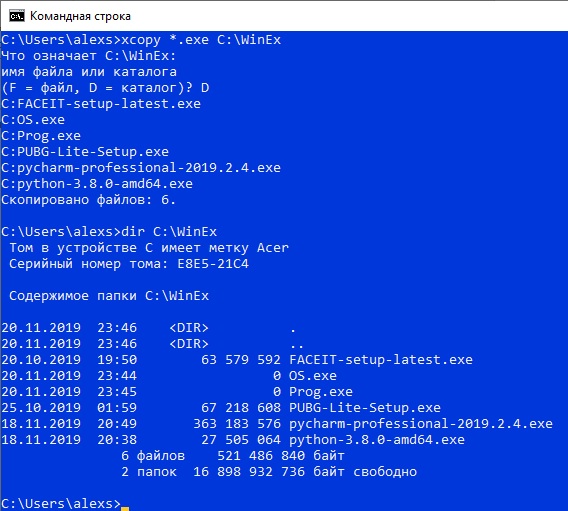


**Students.txt sortStudents.txt**



1. Создал текстовый файл, содержащий справочные сведения по командам DIR, COPY, XCOPY



1. 
2. Скопировал все имеющиеся в Каталоге растровые графические файлы в каталог WinGrafika на диске C:
3. Скопировал все имеющиеся в Каталоге исполняемые файлы в каталог WinEx на диске C:

Контрольные вопросы:

1. **Достоинства и недостатки интерфейса командной строки.**

Достоинства**:**

* Любую команду можно вызвать небольшим количеством нажатий.
* Можно управлять программами, которые не имеют графический интерфейс.
* Просмотрев содержимое консоли, можно повторно увидеть промелькнувшее сообщение, которое не было прочитано.

Недостатки:

* Интерфейс командной строки не является понятным полностью для пользователей, которые начали знакомство с компьютером с графического режима.
* Искать неизвестную команду по справочникам не менее сложно, чем отыскивать в меню нужную команду.
* Ввод некоторых параметров с клавиатуры может быть затруднительным.
* Если в программе должен быть полноценный [скриптовый язык](https://dic.academic.ru/dic.nsf/ruwiki/42896), приходится либо поддерживать два разных языка (консольный и скриптовый), либо отказываться от командной строки в пользу скриптового языка, либо совмещать эти два языка.

**2.** **Инструменты командной строки для автоматизации**

В последних версиях ОС поддерживается несколько стандартных инструментов автоматизации: оболочка командной строки cmd.exe, среда выполнения сценариев Windows Script Host и оболочка Microsoft PowerShell.

**3**. **Настраиваемые свойства интерпретатора**

Цвет, расположение, шрифт, размер курсора, запоминание команд, правка.

**4**. **Различие между внутренними и внешними командами. Примеры внешних и внутренних команд.**

Некоторые команды распознаются и выполняются непосредственно самим командным интерпретатором – такие команды называются внутренними (например, COPY или DIR). Другие команды ОС представляют собой отдельные программы, расположенные по умолчанию в том же каталоге, что и cmd.exe, которые Windows загружает и выполняет аналогично другим программам. Такие команды называются внешними (например, MORE или XCOPY).

**5**. **Структура команды интерпретатора**

Все команды имеют формат записи:

Команда  А1  А2  .../К1/К2/... , где А1, А2, ... – аргументы команды (диски, файлы, каталоги), К1, К2, ... –  параметры или ключи (указывают на особенности выполнения команды).

Параметры и аргументы команды не являются обязательными.

Аргументы отделяются пробелами, а ключи знаком / (косая черта).

Пример: Команда  /?          – справка по команде.

**6**. **Получение информации о конкретной команде**

Для того чтобы получить справочную информацию об определенной команде нужно ввести cmd /?. Например, [ping /?](http://cmd4win.ru/administrirovanie-seti/diagnostika-sety/50-ping) выдаст все возможные параметры команды отправки пакетов на заданный узел.

**7. Групповые символы(шаблоны) и их использование**

При работе в командной строке имена файлов можно заменять групповыми символами, «\*» (звездочкой) и «?» (вопросительным знаком).

Символ «\*» заменяет любое количество символов в названии файла, а символ «?» обозначает присутствие или отсутствие одного символа в названии файла.

Например, запись «\*.txt» обозначает все текстовые файлы.

При использовании имен файлов, состоящих из более чем одного слова, необходимо заключать их в двойные кавычки.

Например, создадим на диске «С» папку «новая папка», а в ней файл «новый документ». Тогда для открытия этого файла, путь к нему необходимо заключить в двойные кавычки.

**8.** **Перенаправление ввода/вывода и конвейеризация команд**

С помощью переназначения устройств ввода/вывода одна программа может направить свой вывод на вход другой или перехватить вывод другой программы, используя его в качестве своих входных данных.

Из командной строки эти возможности реализуются следующим образом.

Для того, чтобы перенаправить текстовые сообщения, выводимые какой-либо командой, в текстовый файл, нужно использовать конструкцию

***команда > имя\_файла***

Если при этом заданный для вывода файл уже существовал, то он перезаписывается (старое содержимое теряется), если не существовал — создается.

Можно также не создавать файл заново, а дописывать информацию, выводимую командой, в конец существующего файла. Для этого команда перенаправления вывода должна быть задана так:

***команда >> имя\_файла***

С помощью символа <  можно прочитать входные данные для заданной команды не с клавиатуры, а из определенного (заранее подготовленного) файла:

***команда < имя\_файла***

**9-10**. **Условное выполнение и группировка команд. Назначение символов &, &&, || и ()**

В командной строке Windows NT/2000/XP можно использовать специальные символы, которые позволяют вводить несколько команд одновременно и управлять работой команд в зависимости от результатов их выполнения. С помощью таких символов условной обработки можно содержание небольшого пакетного файла записать в одной строке и выполнить полученную составную команду.

Используя символ амперсанда &, можно разделить несколько утилит в одной командной строке, при этом они будут выполняться друг за другом.

Условная обработка команд в Windows осуществляется с помощью символов && и || следующим образом. Двойной амперсанд && запускает команду, стоящую за ним в командной строке, только в том случае, если команда, стоящая перед амперсандами была выполнена успешно.

Два символа || осуществляют в командной строке обратное действие.

Несколько утилит можно сгруппировать в командной строке с помощью *круглых скобок*.

**11**. **Команды для работы с файловой системой – названия и возможности**

**Команда CD** – изменение текущего каталога.

**Команда COPY** – копирование одного или нескольких файлов.

**Команда XCOPY** - используется для копирования файлов и каталогов с сохранением их структуры. По сравнению с командой COPY имеет более широкие возможности и является наиболее гибким средством копирования в командной строке Windows.

**Команда DIR** – вывод информации и содержимого дисков и каталогов.

Команды MKDIR и RMDIR – используются для создания нового каталога и удаления уже существующего пустого каталога соответственно.

**Команда DEL** – удаление одного или нескольких файлов.

**Команда REN** – переименование файлов и каталогов.

**Команда MOVE** – перемещение файлов и папок.

**12.** **Достоинства и недостатки команд COPY и XCOPY**

**XCOPY:**

- **Достоинства**: копирует и каталоги, и файлы

- **Недостатки**: может работать только с файлами и каталогами, но не с устройствами

**COPY:**

- **достоинства**: копирует и перемещает файлы

- **недостатки**: копирует только файлы

**13.** **Назначение команды ECHO примеры её использования**

Команда **ECHO** применяется для вывода текстовых сообщений на стандартный вывод и для переключения режима отображения команд на экране.  
  
Примеры использования:  
  
ECHO - при вводе команды без параметров, отображается текущий режим отображения команд:  
  
Режим вывода команд на экран (ECHO) включен.  
  
Для вывода пустой строки, используется команда ECHO с точкой:  
  
echo.  
  
echo Hello World!!! - вывод сообщения Hello World!!! на экран.  
  
Также команда echo используется для создания нового файла:  
  
echo 1 >newfile.cmd - вывести символ "1" в файл newfile.cmd. Если такого файла не существует, он будет создан, если существует, то будет перезаписан.  
  
**14.** **Команда DIR и её возможности**

Команда **DIR**позволяет отобразить список файлов и подкаталогов для указанного каталога. Список может быть отсортирован по множеству критериев, задаваемых параметрами командной строки.

Формат командной строки:   
DIR [диск:][путь][имя\_файла] [/A[[:]атрибуты]] [/B] [/C] [/D] [/L] [/N] [/O[[:]порядок]] [/P] [/Q] [/S] /T[[:]время]] [/W] [/X] [/4]   
  
Параметры командной строки:   
[диск:][путь][имя\_файла] - Диск, каталог и/или файлы, которые следует включить в список.  
  
/A - Вывод файлов с указанными атрибутами:  
D - Каталоги   
R - Доступные только для чтения  
H - Скрытые файлы  
A - Файлы для архивирования  
S - Системные файл  
-S - НЕ системные файлы.  
/B - Вывод только имен файлов.  
  
/C - Применение разделителя групп разрядов для вывода размеров файлов (по умолчанию). Для отключения этого режима служит ключ /-C.  
/D - Вывод списка в несколько столбцов с сортировкой по столбцам.

/L – Использование нижнего регистра для имен файлов.  
/N - Отображение имен файлов в крайнем правом столбце.  
/O - Сортировка списка отображаемых файлов.  
/P - Пауза после заполнения каждого экрана.  
/Q - Вывод сведений о владельце файла.  
/R - Отображение альтернативных потоков файлов.  
/S - Вывод списка файлов из указанного каталога и его подкаталогов.  
/T - Выбор поля времени для отображения и сортировки  
C - Создание (время создания - Create )  
A  - Последнее использование (последний доступ - Accsess )  
W - Последнее изменение (последняя модификация Write)  
/W - Вывод списка в несколько столбцов.  
/X - Отображение коротких имен для файлов, чьи имена не соответствуют стандарту 8.3. Формат аналогичен выводу с ключом /N, но короткие имена файлов выводятся слева от длинных. Если короткого имени у файла нет, вместо него выводятся пробелы.   
/4 - Вывод номера года в четырехзначном формате  
  
Стандартный набор ключей можно записать в переменную среды DIRCMD. Для отмены их действия введите в команде те же ключи с префиксом "-", например: /-W.

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

При создании текстового файла интерпретатор использует кодировку кириллица (DOS). Чтобы получить читаемую твердую копию нужно переназначить вывод в файл с разрешением .txt