

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

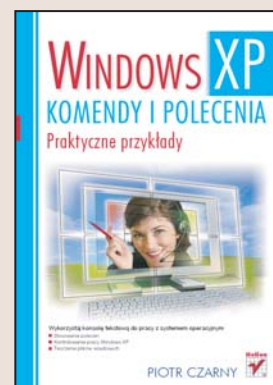
ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

Windows XP. Komendy i polecenia. Praktyczne przykłady

Autor: Piotr Czarny
ISBN: 83-246-0478-2
Format: B5, stron: 320
[Przykłady na ftp: 20 kB](#)



Wykorzystaj konsolę tekstową do pracy z systemem operacyjnym

- Stosowanie poleceń
- Kontrolowanie pracy Windows XP
- Tworzenie plików wsadowych

Współczesne systemy operacyjne przyzwyczyły użytkowników do ikon, okien i rozwijanych menu, za pomocą których można wykonać prawie każdą operację. Nadal jednak istnieje możliwość pracy w trybie tekstowym. Dlaczego? Odpowiednio wykorzystane polecenia trybu tekstowego pozwalają zautomatyzować wiele czynności. Czasami również – na przykład w wypadku awarii systemu i ataku wirusa – użycie konsoli i ręczne wpisanie poleceń jest po prostu jedynym rozwiązaniem.

Dzięki książce „Windows XP. Komendy i polecenia. Praktyczne przykłady” poznasz zasady stosowania poleceń tekstowych w codziennej pracy z systemem Windows XP. Nauczysz się wykorzystywać je do wykonywania operacji na plikach i katalogach, zarządzania zadaniami i procesami systemowymi oraz do odzyskiwania danych. Dowiesz się, w jaki sposób konfigurować i testować połączenia sieciowe, montować nowe woluminy i dzielić dysk na partycje.

- Operacje na katalogach
- Edycja plików tekstowych
- Porównywanie zawartości plików
- Kontrola środowiska pracy
- Montowanie urządzeń systemowych i woluminów
- Ustawianie parametrów ładowania systemu
- Tworzenie kopii zapasowych
- Konfigurowanie parametrów połączeń sieciowych

Przekonaj się, że Windows XP to nie tylko interfejs graficzny



Spis treści

Wstęp	7
Rozdział 1. System operacyjny — FAQ	11
1.1. Co oznaczają skróty DOS i MS-DOS?	11
1.2. Co oznacza skrót XP?	11
1.3. Z jakich elementów składa się system komputerowy?	12
1.4. Co to jest system operacyjny?	13
1.5. Na czym polega przetwarzanie wsadowe?	13
1.6. Co to jest program wsadowy?	14
1.7. Co to jest spooling?	14
1.8. Na czym polega praca z podziałem czasu?	14
1.9. Co to jest swapping?	15
1.10. Kiedy komputery mogą komunikować się ze sobą?	15
1.11. Co to jest protokół sieciowy?	15
1.12. Jakie protokoły sieciowe są zainstalowane w komputerze?	16
Rozdział 2. Tryb tekstowy	17
2.1. Jak pracować w trybie konsoli?	17
2.2. Jak pracować z programami wsadowymi?	19
2.3. MS-DOS i Windows XP	25
2.4. Co można uzyskać, włączając rozszerzenia poleceń?	28
Rozdział 3. Operacje na katalogach	35
3.1. Jak zakładać katalogi?	35
3.2. Jak zmieniać nazwy katalogów?	41
3.3. Jak usuwać katalogi?	43
3.4. Jak dołączać katalogi?	46
3.5. Jak zmieniać katalogi?	47
3.6. Jak wyświetlać graficzną strukturę katalogów?	49
3.7. Jak kopiować drzewo katalogów?	51
Rozdział 4. Operacje na plikach	57
4.1. Jak wyświetlać zawartość pliku tekstowego?	57
4.2. Jak wyszukiwać ciągi znaków?	58
4.3. Jak sortować wiersze w pliku?	63
4.4. Jak drukować pliki tekstowe?	65
4.5. Jak przenosić pliki?	66
4.6. Jak usuwać pliki?	66
4.7. Jak wyświetlić listę plików i katalogów?	69
4.8. Jak zmieniać rozszerzenia nazw plików?	72

4.9. Jak zmieniać atrybuty plików?	73
4.10. Jak ustawiać listę kontroli dostępu do plików?	74
4.11. Jak porównywać pliki?	76
4.12. Jak kopiować pliki?	79
4.13. Jak zmieniać nazwy plików?	81
4.14. Jak zamieniać pliki?	82
4.15. Jak porównywać zawartość dyskietek?	83
4.16. Jak kopiować zawartości dyskietek?	84
4.17. Jak konwertować pliki wykonywalne na binarne?	85
4.18. Jak rozpakowywać pliki instalacyjne?	86
Rozdział 5. Edycja plików	87
5.1. Jak działa edytor tekstowy?	87
5.2. Jak działa edytor liniowy?	92
5.3. Jak wstawiać komentarze?	93
Rozdział 6. Zarządzanie zadaniami	95
6.1. Jak uruchamiać polecenia w określonym czasie?	95
6.2. Jak konfigurować wyzwalacze zdarzeń?	103
6.3. Jak wywoływać programy wsadowe?	107
6.4. Jak wstrzymywać programy wsadowe?	110
6.5. Jak wyłączać przerywanie wykonywania zadań?	111
6.6. Jak wyświetlać nowe okno?	112
6.7. Jak zmienić tytuł okna wiersza poleceń?	113
6.8. Jak wyczyścić okno wiersza poleceń?	114
6.9. Jak testować i edytować narzędzia?	115
6.10. Jak wykonywać zadania w pętli?	115
6.11. Jak przejść do instrukcji oznaczonej etykietą?	118
6.12. Jak napisać instrukcję warunkową?	120
6.13. Jak sterować wyświetlaniem informacji na ekranie?	124
6.14. Jak zachować bieżący katalog dla polecenia POPD?	125
6.15. Jak przejść do katalogu zachowanego przez polecenie PUSHD?	126
6.16. Jak definiować zmienne środowiskowe dla plików wsadowych?	126
6.17. Jak definiować parametry wykonywania programów?	128
6.18. Jak ładować program powyżej pierwszych 64 kB?	129
6.19. Jak zmienić położenie parametrów w pliku wsadowym?	129
6.20. Co to jest host skryptów systemu Windows?	131
6.21. Jak kończyć pracę skryptów wsadowych?	132
6.22. Jak wylogowywać się?	133
6.23. Jak wyszukiwać i odłączać otwarte pliki?	134
6.24. Jak zamknąć system?	138
Rozdział 7. Środowisko	139
7.1. Jak odczytać wersję systemu operacyjnego?	139
7.2. Jak zdefiniować tekst zgłoszenia trybu konsoli?	141
7.3. Jak ustawić ścieżkę wyszukiwania dla plików wykonywalnych?	142
7.4. Jak ustawiać kolor tła i napisów konsoli?	143
7.5. Jak ustawić stronę kodową?	145
7.6. Jak ładować informacje specyficzne dla kraju?	145
7.7. Jak dostosować klawiaturę do wymagań języka?	146
7.8. Jak ustawić datę i czas systemu operacyjnego?	147
7.9. Jakie są dodatkowe funkcje wiersza poleceń?	148
7.10. Jak wyświetlać komunikaty ekranowe?	151
7.11. Jak wprowadzić zmiany środowiska przez plik wsadowy?	153
7.12. Jak przyspieszać operacje dyskowe?	153

7.13. Jak drukować grafikę?	154
7.14. Jak drukować znaki w trybie graficznym?	155
7.15. Jak uzyskiwać informacje o poleceniach systemu Windows XP?	156
7.16. Jak odczytać zajętość pamięci RAM?	156
7.17. Jak konfigurować urządzenia systemowe?	157
7.18. Jak definiować zmienne środowiskowe?	161
7.19. Jak współdzielić zasoby?	165
7.20. Jak włączyć dysk wirtualny?	165
7.21. Jak ustawić parametry ładowania systemu?	166
7.22. Jak wyświetlić listę sterowników urządzeń?	168
7.23. Jak edytować informacje o zdarzeniach?	169
7.24. Jak odświeżać zasady grup?	169
7.25. Jak zarządzać dziennikami zdarzeń?	170
7.26. Jak wyświetlić informacje o uruchomionych procesach?	172
7.27. Jak zamykać procesy?	173
7.28. Zestaw zasad Rop	174
7.29. Ochrona plików systemu Windows	175
7.30. Przywracanie ustawień początkowych	175
7.31. Uruchamianie poleceń	176
Rozdział 8. System plików	179
8.1. Jak wykonywać kopie zapasowe?	179
8.2. Jak partycjonować dysk?	180
8.3. Jak formatować nośniki?	195
8.4. Jak montować woluminy?	197
8.5. Jak odczytać dane woluminu?	198
8.6. Jak ustawić etykietę woluminu lub dysku?	198
8.7. Jak kontrolować poprawności zapisywania plików na dysku?	199
8.8. Jak defragmentować woluminy?	199
8.9. Jak testować i naprawiać nośniki?	200
8.10. Jak odzyskiwać dane z uszkodzonego dysku?	202
8.11. Jak kompresować dane na partycjach NTFS?	202
8.12. Jak konwertować system FAT na NTFS?	204
8.13. Jak szyfrować dane na partycjach NTFS?	205
8.14. Jak mierzyć wydajności dysków?	206
8.15. Zarządzanie przydziałami dysku z wiersza polecenia	207
Rozdział 9. Praca w sieci	221
9.1. Jak odczytać adres MAC karty sieciowej?	221
9.2. Jak skonfigurować kartę sieciową?	223
9.3. Jak ustawić parametry połączenia sieciowego?	226
9.4. Jak testować połączenie sieciowe?	251
9.5. Jak odczytać statystykę połączenia TCP/IP?	252
9.6. Jak wysyłać komunikaty do użytkowników?	254
9.7. Jak pracować z terminalem?	255
9.8. Jak przysyłać pliki?	256
9.9. Jak można wyświetlić tabelę translacji adresów IP?	258
9.10. Jak monitorować połączenia zarejestrowane przez Menedżera wywołań ATM?	259
9.11. Wyświetlanie stanu kolejki wydruku	260
9.12. Line Printer Daemon	261
9.13. Statystyki protokołu i bieżące połączenia TCP/IP	262
9.14. Modyfikowanie konfiguracji sieciowej uruchomionego komputera	263
9.15. Polecenia netsh dotyczące zakresów serwera DHCP	281
9.16. Netsh — zastosowania praktyczne	289

Dodatek A Parametry wsadowe, zmienne i filtry	293
A.1. Parametry wsadowe	293
A.2. Modyfikatory	295
A.3. Stosowanie wielu poleceń i symboli przetwarzania warunkowego	296
A.4. Systemowe i lokalne zmienne środowiskowe	297
A.5. Ustawianie zmiennych środowiskowych	300
A.6. Podstawianie wartości zmiennych środowiskowych	300
A.7. Filtry	301
A.8. Polecenie find	303
A.9. Polecenie sort	304
Skorowidz	307

Rozdział 3.

Operacje na katalogach

Przez analogię do pracy w biurze katalogi można porównać do segregatorów, a pliki do dokumentów. Aby zaprowadzić porządek w biurze, należy dokumenty pogrupować i wpiąć do segregatorów. W przypadku plików trzeba utworzyć strukturę katalogów ułatwiającą zlokalizowanie danego pliku i zapisywać pliki w przeznaczonych do tego celu miejscach.

3.1. Jak zakładać katalogi?

Katalogi można zakładać za pomocą dwóch poleceń: `md` oraz `mkdir`.

md

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `md` umożliwia założenie katalogu.

Ma ono następującą składnię:

```
MD [dysk:]ścieżka
```

gdzie:

dysk — litera napędu;

ścieżka — ścieżka dostępu do zakładanego katalogu.

Przykład 3.1.

Utwórz na dysku bieżącym folder *a*.

Wpisz w oknie konsoli polecenie:

```
md a
```

Przykład 3.2.

Utwórz folder *a* na dysku o nazwie *D*.

Wpisz w oknie konsoli polecenie:

```
md d:\a
```

Przykład 3.3.

Utwórz na dysku o nazwie *C* folder *a*. W folderze *a* utwórz folder *b*. W folderze *b* utwórz folder *c*. W folderze *c* utwórz folder *d*.

Wpisz w oknie konsoli polecenie:

```
md c:\a\b\c\d
```

Zapisanie w jednej linii polecenia umożliwiającego utworzenie wielu folderów jest możliwe przy włączonych rozszerzeniach poleceń — domyślna opcja pracy konsoli.

Ostatni przykład dobitnie pokazuje, że możliwości, jakie stwarzają polecenia dostępne w trybie tekstowym, są większe niż oferowane przez interfejs graficzny. Poleceniem wpisanym w jednej linijce można utworzyć wiele katalogów. Osiągnięcie tego samego rezultatu poprzez posługiwanie się „okienkami” jest bardziej pracochłonne.

Na tym przykładzie widać również wady trybu tekstowego. Nowo utworzone foldery nie są widoczne. W trybie graficznym — są widoczne zaraz po ich utworzeniu.

Przykład 3.4.

Uruchom tryb konsoli z wyłączonymi rozszerzeniami poleceń. Utwórz na dysku o nazwie *C* folder *a*. W folderze *a* utwórz folder *b*. W folderze *b* utwórz folder *c*. W folderze *c* utwórz folder *d*.



Do uruchomienia konsoli z wyłączonymi rozszerzeniami poleceń wykorzystaj plik wsadowy *bezrozszt.bat*.

1. Wpisz w oknie konsoli polecenie `md c:\a\b\c\d`. Polecenie nie zostało wykonane (rysunek 3.1). Powodem tego są wyłączone rozszerzenia.
2. Identyczną strukturę katalogów, jaką chcieliśmy utworzyć w punkcie 1., można założyć pojedynczymi poleceniami. Wpisz polecenie `md a`. Utworzony został katalog *a*.
3. Przejdź do katalogu *a*, wpisując polecenie `cd a`.
4. Wpisz polecenie `md b`. Utworzony został katalog *b*.
5. Przejdź do katalogu *b*, wpisując polecenie `cd b`.
6. Wpisz polecenie `md c`. Utworzony został katalog *c*.

7. Przejdź do katalogu *c*, wpisując polecenie `cd c`.
8. Wpisz polecenie `md d`. Utworzony został katalog *d*.
9. Przejdź do katalogu *d*, wpisując polecenie `cd d`. Uzyskaliśmy taką samą strukturę katalogów jak w przykładzie 3.3. Wymagało to jednak wpisania dużo większej liczby poleceń.

Rysunek 3.1.

*Po wyłączeniu
rozszerzeń
utworzenie struktury
podkatalogów
wymaga wydania
wielu poleceń*

```

C:\WINDOWS\System32\cmd.exe
C:\Test>cmd /e:off
Microsoft Windows XP [Wersja 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Test>cd\
C:\>md c:\a\b\c\d
Nie można utworzyć katalogu lub pliku.
C:\>md a
C:\>cd a
C:\a>md b
C:\a>cd b
C:\a\b>md c
C:\a\b>cd c
C:\a\b\c>md d
C:\a\b\c>cd d
C:\a\b\c\d>_

```

Przykład 3.5.

Napisz program, który będzie zakładał folder o nazwie identycznej z zapisem bieżącej daty.

Polecenie jest wykonywane przez program:

```

:: datedir.bat
:: Program zakłada katalog o nazwie identycznej jak bieżąca data
@ECHO OFF
MD %DATE%

```

Pora na wyjaśnienie, co i jak wykonuje program.

Pierwsza linia rozpoczyna się od dwukropka. Oznacza on, że wszystko, co znajduje się na prawo od niego, powinno być traktowane jako komentarz. Nie jest to instrukcja dla komputera, ale informacja dla człowieka, który będzie używał programu. Umieszczenie komentarzy w programach jest dobrym zwyczajem, gdyż zamiast śledzić treść programu, by się dowiedzieć, do czego on służy — wystarczy przeczytać komentarz. Drugi dwukropek ułatwia odnalezienie komentarza w długim programie. Po dwukropkach wpisana jest nazwa programu.

Druga linia również zawiera komentarz. Opisana jest w nim funkcja, którą realizuje program.

Linia

```
@ECHO OFF
```

rozpoczyna się od znaku @. Zapobiega on wyświetleniu na początku polecenia pojedynczego wiersza z pliku wsadowego.

Polecenie *echo off* powoduje wyłączenie wyświetlania na ekranie poleceń wydawanych komputerowi przez program.

Linia

```
MD %DATE%
```

zawiera polecenie założenia folderu o nazwie identycznej ze zmienną %DATE%. Zmiennej %DATE% przypisywana jest bieżąca data. Folder zakładany jest w katalogu bieżącym.

Przy włączonych rozszerzeniach poleceń istnieje kilka dynamicznych zmiennych środowiskowych, które są obliczane dynamicznie za każdym razem, gdy wartość zmiennej jest rozwijana:

%CD% — powoduje przypisanie zmiennej ciągu katalogu bieżącego;

%DATE% — powoduje przypisanie zmiennej aktualnej daty;

%TIME% — powoduje przypisanie zmiennej bieżącego czasu;

%RANDOM% — powoduje przypisanie zmiennej losowej liczby dziesiętnej z przedziału od 0 do 32767;

%ERRORLEVEL% — powoduje przypisanie zmiennej wartości ERRORLEVEL;

%CMDEXTVERSION% — powoduje przypisanie zmiennej wersji rozszerzeń procesora poleceń;

%CMDCMDLINE% — powoduje przypisanie zmiennej oryginalnego wiersza polecenia wywołanego przez procesor poleceń.



Więcej informacji na temat polecenia *set* możesz uzyskać, wpisując w oknie konsoli polecenie *set /?*.



Kolejne uruchomienie programu *datedir.bat* nie powoduje zastąpienia istniejącego już folderu.

Przykład 3.6.

Napisz program, który będzie zakładał 3 foldery o nazwach odpowiadających liczbom losowym.

Polecenie jest wykonywane przez program:

```
:: chaos.bat  
:: Program zakłada 3 katalogi o nazwach odpowiadających losowym liczbom.  
@ECHO OFF
```

```
MD %RANDOM%  
MD %RANDOM%  
MD %RANDOM%
```

O ile program z przykładu 3.5 umożliwił zaprowadzenie porządku na dysku twardym, o tyle program *chaos.bat* może wywołać odwrotny skutek. Jeżeli instrukcja MD %RANDOM% będzie uruchamiana w pętli, wówczas program może np. zapełnić dyskietkę folderami o losowych nazwach.

Działanie programu jest bardzo podobne do opisanego w przykładzie 3.5. Różnica polega na użyciu funkcji RANDOM zamiast DATE.

Przykład 3.7.

Napisz program, który będzie zakładał folder o nazwie podanej po wyświetleniu pytania.

Polecenie jest wykonywane przez program:

```
:: kat.bat  
:: Program zakłada katalog o nazwie wpisanej w odpowiedzi na pytanie  
  
@echo off  
set /P kat="Wpisz nazwe katalogu:"  
if exist %kat% echo Katalog %kat% już istnieje! & goto :END  
MD %kat%  
:END
```

Pora na wyjaśnienie, co i jak wykonuje program.

Pierwsza linia rozpoczyna się od dwukropka. Oznacza on, że wszystko, co znajduje się na prawo od niego, powinno być traktowane jako komentarz. Drugi dwukropek ułatwia odnalezienie komentarza w długim programie. Po dwukropkach wpisana jest nazwa programu.

Druga linia również rozpoczyna się od dwóch dwukropków. Po niej zapisany jest opis programu.

Linia

```
@echo off
```

rozpoczyna się od znaku @. Zapobiega on wyświetleniu na początku polecenia pojedynczego wiersza z pliku wsadowego.

Polecenie echo off powoduje wyłączenie wyświetlania na ekranie poleceń wydawanych komputerowi przez program.

Linia

```
set /P kat="Wpisz nazwe katalogu:"
```

zawiera polecenie set. Ma ono następującą składnię:

```
SET /P zmienna=[ciąg_monitu]
```

Przełącznik /P pozwala przypisać zmiennej ciąg znaków wprowadzony przez użytkownika. Przed odczytaniem ciągu znaków wyświetlany jest komunikat.



Ciąg monitu w poleceniu może być pusty.

W analizowanym programie odczytanie powoduje wyświetlenie polecenia wpisania nazwy katalogu, a po wpisaniu tej nazwy — przypisanie zmiennej `kat`.

Polecenie

```
if exist %kat% echo Katalog %kat% już istnieje! & goto :END
```

powoduje sprawdzenie, czy istnieje już folder o takiej nazwie, jaka przypisana została zmiennej `kat`. Jeżeli folder istnieje, pojawia się komunikat i wykonywany jest skok do etykiety `:END`. Etykieta `:END` znajduje się na końcu programu. Skok do niej oznacza zakończenie wykonywania programu.

Jeżeli wprowadzona nazwa folderu jeszcze nie istnieje, wykonywana jest instrukcja

```
MD %kat%
```

Rezultatem jej wykonania jest założenie katalogu o nazwie przypisanej zmiennej `%kat%`.



Zmiennej `kat` można przypisać nie tylko nazwę pojedynczego katalogu, ale i ścieżkę dostępu.

mkdir

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `mkdir` umożliwia założenie katalogu.

Ma ono następującą składnię:

```
MKDIR [dysk:]ścieżka
```

gdzie:

dysk — litera napędu.

ścieżka — ścieżka dostępu do zakładanego katalogu.



Przy włączonych rozszerzeniach poleceń polecenie `MKDIR` tworzy wszystkie pośrednie katalogi w ścieżce.

Przykłady są takie same jak w przypadku polecenia `md`. Inna jest tylko nazwa polecenia.

3.2. Jak zmieniać nazwy katalogów?

Zmiana nazwy katalogu jest czynnością identyczną z przeniesieniem zawartości folderu do folderu o innej nazwie.

move

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `move` przenosi pliki oraz zmienia nazwy plików i katalogów.

Ma ono następującą składnię:

- ♦ przeniesienie jednego lub więcej plików:

```
MOVE [/Y | /-Y] [dysk:][ścieżka]nazwa_pliku1[,...] cel
```

- ♦ zmiana nazwy katalogu:

```
MOVE [/Y | /-Y] [dysk:][ścieżka]katalog1 katalog2
```

gdzie:

`[dysk:][ścieżka]nazwa_pliku1` — ścieżka i nazwy plików do przeniesienia.

`cel` — nowa lokalizacja plików; może być oznaczona literą dysku z dwukropkiem, nazwy katalogu lub ich kombinacją.



Jeżeli przenoszony jest tylko jeden plik, w poleceniu można podać jego nową nazwę.

`[dysk:][ścieżka]katalog1` — nazwa katalogu, którą chcesz zmienić.

`katalog2` — nowa nazwa katalogu.

`/Y` — wyłącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.

`/-Y` — włącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.

Przykład 3.8.

Zmień nazwę pliku `aaa.txt` znajdującego się na dysku `A` na `spis.txt`.

Aby wykonać polecenie:

1. Umieść w napędzie `A` dyskietkę, na której znajduje się plik `aaa.txt`.
2. Uruchom konsolę.

3. Wpisz z klawiatury `a:.`. Aktywnym napędem został napęd *A*.
4. Wpisz z klawiatury `move aaa.txt spis.txt`. Nazwa pliku została zmieniona.

Przykład 3.9.

Przenieś wszystkie pliki o rozszerzeniu `*.txt` z katalogu głównego dyskiety umieszczonej w napędzie *A* do folderu `c:\tekst`. Zastąp bez potwierdzania wszystkie pliki.

Aby wykonać polecenie:

1. Umieść w napędzie *A* dyskietkę, na której znajduje się plik `aaa.txt`.
2. Uruchom konsolę.
3. Wpisz z klawiatury `move /Y a:*.txt c:\tekst\`.

Przykład 3.10.

Napisz program, który będzie przenosił pliki pomiędzy dwoma folderami. Nazwy folderów: źródłowego i docelowego będą parametrami wywołania programu.



Zmienne oznaczane są w plikach wsadowych cyframi od 0 do 9.

1. Zapisz w pliku wsadowym `przenies.bat` program:

```

:: przenies.bat
:: Program przenosi pliki z katalogu źródłowego do docelowego

@echo off
cls
if %2X==X goto pomoc
move %1 %2
Echo Pliki %1 zostaną przeniesione do %2
goto koniec
Rem ---
: pomoc
Echo Program powinien być wywołany z dwoma parametrami
<pliki do przeniesienia><katalog docelowy>
Rem ---
: koniec
Echo on

```

Polecenie znajdujące się w trzeciej linii (`@echo off`) wyłącza echo. Od tej pory na ekranie nie będą wyświetlane wydawane polecenia.

Polecenie `cls` powoduje usunięcie wszelkich napisów z okna konsoli.

Polecenie `if %2X==X goto pomoc` zawiera instrukcję warunkową. Sprawdza ona, czy pierwszy parametr i drugi parametr wywołania programu są identyczne. Jeśli tak — wykonywany jest skok do etykiety `:pomoc`.

Jeżeli pierwszy i drugi parametr wywołania programu są różne, wykonywana jest instrukcja `move %1 %2`. Zmiennej `%1` przypisana jest nazwa obiektu

źródłowego. Zmiennej %2 przypisana jest nazwa obiektu docelowego. Parametry te zostały pobrane z wywołania programu.

Wykonanie polecenia zapisanego w linii Echo Pliki %1 zostaną przeniesione do %2 powoduje wyświetlenie komunikatu o kopiowaniu plików.

Wyświetlenie komunikatu jest ostatnią czynnością wykonywaną przez program w części związanej z przenoszeniem plików. Po wykonaniu kopiowania i wyświetleniu komunikatu wykonywana jest instrukcja goto koniec. Jej rezultatem jest skok do etykiety koniec.

Linia Rem --- zawiera komentarz. W tym przypadku trzy kreski oddzielają graficznie część programu z komunikatem o błędzie od reszty kodu.

Linie rozpoczynające się od znaku : zawierają etykiety. Są to wyróżnione miejsca w programie, do których wykonywane są skoki. W programie są dwie etykiety: pomoc i koniec.

Po etykiecie : pomoc znajduje się tekst komunikatu: Echo Program powinien być wywołany z dwoma parametrami <pliki do przeniesienia><katalog docelowy>. Wyświetlany jest on tylko wtedy, gdy plik źródłowy i docelowy są identyczne.

Ostatnim poleceniem jest Echo on. Powoduje ono ponowne włączenie echa.

2. Testowanie programu wykonamy na dwóch folderach. Utwórz foldery o nazwach: *moje* i *nasze*. W folderze *moje* umieść plik *notatka.txt*.
3. Umieść plik wsadowy z programem w katalogu głównym.
4. Wpisz w oknie konsoli polecenie `przenieś.bat moje nasze`.
5. Sprawdź, czy plik *notatka.txt* został usunięty z folderu *moje* i pojawił się w folderze *nasze*.

3.3. Jak usuwać katalogi?

Jeżeli piszesz program, który do działania wymaga utworzenia folderów roboczych, przed zakończeniem jego pracy należy foldery usunąć. Istnieje instrukcja, która pozwala na usunięcie całego drzewa katalogów.

rd

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie rd umożliwia usuwanie katalogów.

Ma ono następującą składnię:

```
RD [/S] [/Q] [dysk:]ścieżka
```

gdzie:

/S — usuwa określony katalog wraz ze wszystkimi znajdującymi się w nim katalogami i plikami.

/Q — nie jest wymagane potwierdzenie usuwania plików i folderów.



Identyczne działanie i składnię ma polecenie `rmdir`

Przykład 3.11.

Usuń z dysku umieszczonego w napędzie *A* folder *linux* wraz z zawartością. Operacja ma nie wymagać potwierdzenia.

Zadana operacja zostanie wykonana po wpisaniu w oknie konsoli polecenia:

```
rd /s /q a:\linux\
```

Przykład 3.12.

Napisz program, który będzie usuwał z dyskietek umieszczonych w napędzie *A* wszystkie pliki i foldery wraz z zawartością. Operacja ma nie wymagać potwierdzenia.

Zapisz w pliku wsadowym *usun.bat* program:

```
:: usun.bat
:: Program usuwa wszystkie dane z dyskietki umieszczonej w napędzie A
@echo off
cls
Echo Program usunie wszystkie dane z dyskietki umieszczonej w napędzie A
Pause
rd /s /q a:\
Echo on
```

Pierwsze dwie linie programu zawierają jego nazwę i opis.

Polecenie

```
@echo off
```

powoduje wyłączenie wyświetlania na ekranie linii programu. Włączenie wyświetlania wykonywanych poleceń jest realizowane przez ostatnią linię programu.

Polecenie

```
cls
```

ma za zadanie usunięcie z okna konsoli wszystkich napisów.

Polecenie

```
Echo Program usunie wszystkie dane z dyskietki umieszczonej w napędzie A
```

powoduje wyświetlenie na ekranie komunikatu, który rozpoczyna się od słów *Program usunie....*

Polecenie

```
Pause
```

powoduje wyświetlenie na ekranie polecenia Aby kontynuować, naciśnij dowolny klawisz.... Dyskietka nie jest jeszcze czyszczona. Usuwanie jej zawartości rozpocznie się po naciśnięciu klawisza.

Polecenie

```
rd /s /q a:\
```

powoduje usunięcie wszystkich plików i katalogów znajdujących się na dyskietce umieszczonej w napędzie *A*. Przełączniki powodują, że usuwanie będzie wykonywane wraz z folderami (*/s*) oraz nie będzie wymagało potwierdzenia (*/q*).



Program nieodwracalnie oczyszcza nośnik umieszczony w napędzie. Należy zwrócić uwagę, by przez przypadek nie usunąć ewentualnych ważnych danych.

Przykład 3.13.

Napisz program, który będzie usuwał z dyskietek kolejno umieszczanych w napędzie *A* wszystkie pliki i foldery wraz z zawartością. Operacja ma nie wymagać potwierdzenia.

Zapisz w pliku wsadowym *usuwacz.bat* program:

```
:: usuwacz.bat
:: Program usuwa wszystkie dane z dyskietki umieszczonej w napędzie A
@echo off
: start
cls
Echo Włóż dyskietkę do napędu.
pause
cls
rd /s /q a:\
Echo Dyskietka została wyczyszczona.
pause
goto start
```

Program *usuwacz.bat* jest podobny do *usun.bat*. Zasadnicza różnica polega na tym, że *usuwacz.bat* działa w pętli. W czwartej linii widoczna jest etykieta `: start`. Do niej wykonywany jest skok z ostatniej linii programu. Za wykonanie skoku odpowiedzialna jest instrukcja `goto start`.

Po wyczyszczeniu jednej dyskietki program jest gotowy do wyczyszczenia kolejnej. Nie trzeba go ponownie uruchamiać. Aby przerwać działanie programu *usuwacz.bat* należy nacisnąć klawisze *Ctrl+C*.

3.4. Jak dołączać katalogi?

W systemie operacyjnym są lokalizacje „uprzywilejowane”. Po wydaniu polecenia to właśnie w nich poszukiwany jest wywoływany program. Programy znajdujące się w innych lokalizacjach nie są wykonywane.

Poszukiwanie wywołanych plików jest realizowane w następującej kolejności:

- ◆ katalog wyszczególniony w poleceniu,
- ◆ katalog bieżący,
- ◆ katalogi podane w poleceniu append.

append

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `append` pozwala na zdefiniowanie ścieżek dostępu. Jest ono uzupełnieniem polecenia `path`.

Po załadowaniu programu `append` można go wywoływać, podając listę ścieżek dostępu rozdzielonych średnikami. Dzięki temu np. programy mogą otwierać pliki danych tak, jak w bieżącym katalogu.

Ma on następującą składnię:

```
APPEND [[dysk:]ścieżka[:...]] [/X[:ON | :OFF]] [/PATH:ON | /PATH:OFF] [/E]
```

gdzie:

[*dysk:*] *ścieżka* — ścieżka dostępu do dołączanych zasobów.

/X:ON — dołączone katalogi będą uwzględniane przy poszukiwaniu plików i uruchamianiu aplikacji.

/X:OFF — dołączone katalogi będą uwzględniane tylko przy żądaniach otwarcia pliku (ustawienie domyślne).

/PATH:ON — dołączone katalogi będą uwzględniane przy żądaniach plików, dla których już określono ścieżkę (ustawienie domyślne).

/PATH:OFF — anulowanie opcji */PATH:ON*.

/E — kopia listy dołączonych katalogów będzie przechowywana w zmiennej środowiskowej `APPEND`. Przełącznik */E* może być używany tylko przy pierwszym wywołaniu polecenia `APPEND` po uruchomieniu systemu.



Wydanie polecenia `append` (bez opcji) powoduje pokazanie aktualnej ścieżki dostępu.



Wydanie polecenia `append ;` (ze średnikiem) powoduje usunięcie dołączonych ścieżek.

Przykład 3.14.

Dołącz katalog *XP* znajdujący się na dysku *C*, aby był przeglądany przy poszukiwaniu plików i uruchamianiu aplikacji.

Zadana operacja zostanie wykonana po wydaniu polecenia:

```
append c:\xp /x:off
```

Przykład 3.15.

Wyświetl listę dołączonych katalogów.

Zadana operacja zostanie wykonana po wydaniu polecenia:

```
append <Enter>
```

Przykład 3.16.

Wyczyść listę dołączonych katalogów.

Zadana operacja zostanie wykonana po wydaniu polecenia:

```
append ;<Enter>
```

3.5. Jak zmieniać katalogi?

cd

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `cd` wyświetla nazwę bieżącego katalogu lub powoduje jego zmianę.

Ma ono następującą składnię:

```
CD [/D] [dysk:][ścieżka]
```

gdzie:

[*dysk:*] [*ścieżka*] — ścieżka dostępu.

/D — wraz ze zmianą bieżącego katalogu na dysku zmieniany będzie bieżący dysk.

Polecenie ma kilka wyjątków:

`CD ..` — przejście do katalogu nadrzędnego.

CD \ — przejście do katalogu głównego.

CD *dysk*: — wyświetlenie bieżącego katalogu na wskazanym dysku.

CD — bieżący dysk i katalog.

Przykład 3.17.

Przejdź do katalogu *temp*, który znajduje się na dysku C.

Zadana operacja zostanie wykonana po wydaniu polecenia:

```
cd c:\temp<Enter>
```

Przykład 3.18.

Przejdź do katalogu nadrzędnego.

Zadana operacja zostanie wykonana po wydaniu polecenia:

```
cd..<Enter>
```



Po włączeniu rozszerzeń poleceń ciąg katalogu bieżącego jest konwertowany w celu używania ciągu takiego jak nazwy dyskowe.



Po włączeniu rozszerzeń polecenie cd nie traktuje spacji jako ograniczników. Można go używać w połączeniu z nazwą podkatalogu, która zawiera spację bez otaczających nazwę cudzysłowów.

Przykład 3.19.

Napisz program, który będzie uruchamiał arkusz kalkulacyjny *Lotus* wraz z plikiem; plik zostanie podany jako parametr polecenia.

Zapisz w pliku wsadowym *lot.bat* program:

```
:: lot.bat
:: Program uruchamia arkusz Lotus oraz plik, którego nazwa
:: jest parametrem wywołania programu lot.bat
@echo off
C:
CD\LOTUS\123
123 %1
CLR
```

W pierwszych trzech liniach znajdują się: nazwa programu i jego opis.

Polecenie

```
@echo off
```

powoduje wyłączenie wyświetlania w oknie konsoli poleceń wydawanych przez program.



W przykładzie założyłem, że arkusz *Lotus* został zainstalowany w folderze, do którego ścieżka dostępu ma postać `c:\lotus\123`.

W linii piątej następuje zmiana aktywnego dysku na *C*. W kolejnej linii znajduje się polecenie przejścia z katalogu głównego do folderu *123*, który znajduje się w folderze *Lotus*.

Polecenie

```
123 %1
```

powoduje uruchomienie programu *123* z arkuszem, którego nazwa zostanie podana jak parametr wywołania programu *lot.bat*.

Ostatnia instrukcja wykonywana jest po zakończeniu pracy z arkuszem. Powoduje ona wyczyszczenie okna konsoli.

chdir

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `chdir` ma takie same zastosowanie i składnię jak `cd`. Różni się tylko nazwą polecenia.

3.6. Jak wyświetlać graficzną strukturę katalogów?

To nie pomyłka — posługując się poleceniami działającymi w trybie tekstowym można wyświetlać graficzną strukturę katalogów. Wzajemne zależności pomiędzy nimi sygnalizowane są za pomocą kresek.

tree

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `tree` przedstawia graficznie strukturę folderów na dysku lub w określonej ścieżce.

Ma ono następującą składnię:

```
TREE [dysk:][ścieżka] [/F] [/A]
```

gdzie:

[*dysk*:][*ścieżka*] — ścieżka dostępu do miejsca na dysku, którego struktura ma zostać wyświetlona.

/F — wyświetla strukturę folderów oraz zawarte w nich pliki.

/A — rysuje strukturę katalogów, używając znaków *ASCII* zamiast znaków rozszerzonych.

Przykład 3.20.

Wyświetl strukturę katalogów utworzonych na dyskiecie *A*.

Aby wykonać polecenie:

1. Włóż dyskietkę do napędu oznaczonego symbolem *A*.
2. Uruchom konsolę i wpisz z klawiatury `tree a:`.

Przykład 3.21.

Wyświetl strukturę katalogów utworzonych na dyskiecie *A* i zawarte w nich pliki.

Aby wykonać polecenie:

1. Włóż dyskietkę do napędu oznaczonego symbolem *A*.
2. Uruchom konsolę i wpisz z klawiatury `tree a: /f`.

Przykład 3.22.

Napisz program o nazwie *spis.txt*, który będzie katalogował zawartość nośnika włożonego do napędu i zapisywał strukturę folderów i plików.

Zapisz w pliku wsadowym *spis.bat* program:

```
:: spis.bat
:: Program kataloguje zawartość nośnika
:: umieszczonego w napędzie, którego nazwa
:: jest parametrem wywołania programu spis.bat

@echo off
tree %1: /f /a >>c:\spis.txt.
```

W pierwszych czterech liniach znajdują się: nazwa programu i jego opis.

Polecenie

```
@echo off
```

powoduje wyłączenie wyświetlania w oknie konsoli poleceń wydawanych przez program.

Polecenie

```
tree %1: /f /a >>c:\spis.txt.
```

powoduje wygenerowanie drzewa plików i folderów. Literowe oznaczenie napędu jest parametrem wywołania programu *spis.bat*. Drzewo zostaje narysowane z wykorzystaniem znaków *ASCII* zamiast znaków rozszerzonych. Plik ze spisem zostaje zapisany w katalogu głównym.



Znak >> powoduje przekierowanie strumienia danych do pliku. Jeśli plik o określonej nazwie nie istnieje — zostanie utworzony.



Znak > powoduje dopisanie strumienia danych na końcu określonego pliku.

3.7. Jak kopiować drzewo katalogów?

Kopiowanie drzew katalogów umożliwia wykonanie kopii struktury folderów wraz z ich zawartością.

xcopy

Polecenie systemów: *Windows XP Home Edition* i *Windows XP Professional*

Polecenie `xcopy` umożliwia kopiowanie plików i drzew katalogów. Wywołane bez podania źródła kopiuje pliki z bieżącego katalogu. Jeśli podane jako miejsca przeznaczenia katalogi nie istnieją, polecenie `XCOPY` tworzy je.

Ma ono następującą składnię:

```
XCOPY źródło [ce1] [/A | /M] [/D[:data]] [/P] [/S [/E]] [/V] [/W] [/C] [/I] [/Q]
[/F] [/L] [/G] [/H] [/R] [/T] [/U] [/K] [/N] [/O] [/X] [/Y] [/Y] [/Z]
[/EXCLUDE:plik1[+plik2][+plik3]...]
```

gdzie:

źródło — plik(i) przeznaczone do skopiowania.

ce1 — miejsce, gdzie pliki mają zostać skopiowane.

`/A` — kopiuje tylko pliki z ustawionym atrybutem archiwizowania, nie zmieniając atrybutu.

`/M` — kopiuje pliki z ustawionym atrybutem archiwalnym, wyłączając ten atrybut. Opcja pozwala na korzystanie z polecenia `XCOPY` zamiast `BACKUP` w celu wykonania ograniczonej operacji archiwizowania.

`/D:m-d-r` — kopiuje pliki zmienione określonego dnia lub później.



Jeżeli data nie została podana, kopiowane są tylko pliki nowsze niż docelowe.

`/EXCLUDE:plik1[+plik2][+plik3]...` — określa listę plików zawierających ciągi. Każdy ciąg powinien być zapisany w osobnym wierszu w plikach. Jeżeli jakiś ciąg pasuje do dowolnej części ścieżki absolutnej kopiowanego pliku — plik ten nie zostanie skopiowany.

- /P — kopiowanie każdego pliku wymaga każdorazowego potwierdzenia.
- /S — kopiuje katalogi i podkatalogi z pominięciem pustych. Katalogi, które nie istnieją na dysku przeznaczenia, są tworzone.
- /E — kopiuje wszystkie podkatalogi źródła, nawet jeżeli są puste. Katalogi nieistniejące na dysku przeznaczenia są tworzone.
- /V — włączenie weryfikacji nowych plików.
- /W — włączenie monitowania o naciśnięcie klawisza przed kopiowaniem.
- /C — kopiowanie będzie kontynuowane nawet po wystąpieniu błędu.
- /I — jeśli „cel” nie istnieje, a kopiowany jest więcej niż jeden plik, zakłada się, że „cel” musi być katalogiem.
- /Q — podczas kopiowania nie są wyświetlane nazwy kopiowanych plików.
- /F — podczas kopiowania wyświetlane są pełne nazwy plików źródłowych i docelowych.
- /L — listowanie plików przeznaczonych do skopiowania.
- /G — kopiowane będą również pliki szyfrowane, nawet gdy lokalizacja docelowa nie obsługuje szyfrowania.
- /H — kopiowane będą pliki ukryte i systemowe.
- /R — zastępowane będą pliki z atrybutem tylko do odczytu.
- /T — generowana jest struktura katalogów. Pliki nie są kopiowane. Pomijane są puste katalogi i podkatalogi.
- /U — kopiowane są tylko te pliki, które istnieją już w miejscu docelowym.
- /K — kopiowane są tylko atrybuty.



Xcopy resetuje atrybut *Tylko do odczytu*.

- /N — kopiowanie z użyciem krótkich nazw.
- /O — kopiowanie informacji o właścicielu i listy *ACL*.
- /X — kopiowanie ustawień inspekcji plików (implikuje użycie /O).
- /Y — wyłącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.



Przełącznik /Y można umieścić w zmiennej środowiskowej COPYCMD.

- /-Y — włącza monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego.
- /Z — kopiuje pliki w trybie umożliwiającym ponowne uruchomienie.

Przykład 3.23.

Napisz program wsadowy, który będzie kopiował zawartość folderu *c:\moje* na dyskietkę umieszczoną w napędzie oznaczonym literą *A*.

1. Napisz program:

```
:: xcopy1.bat
:: Program kopiuje zawartość folderu c:\moje na dyskietkę umieszczoną
w napędzie a
@Echo off
cls
cd c:\moje
Echo Umieść dyskietkę w stacji A:
pause
xcopy *.* A: /e
Echo Kopiowanie zakończone...
```

Polecenie `@Echo off` wyłącza wyświetlanie na ekranie poszczególnych linii programu. Wyświetlane są komunikaty wywołane wykonywaniem poleceń.

Polecenie `cls` czyści ekran konsoli.

Polecenie `cd c:\moje` powoduje, że katalogiem bieżącym jest *c:\moje*.

Polecenie `Echo Umieść dyskietkę w stacji A:` powoduje wyświetlenie na ekranie polecenia `Umieść dyskietkę w stacji A:`.

Polecenie `pause` powoduje wyświetlenie na ekranie polecenia `Aby kontynuować, naciśnij dowolny klawisz...`. Wykonywanie kolejnych linii programu zostaje wstrzymane aż do czasu naciśnięcia klawisza. W połączeniu z poprzednią linią daje to następujący efekt: wyświetlane jest polecenie włożenia dyskietki do napędu — program czeka tak długo, aż naciśnięty zostanie klawisz.

Polecenie `xcopy *.* A: /e` powoduje skopiowanie z katalogu bieżącego na nośnik umieszczony w napędzie *A:* wszystkich plików i folderów. Użycie przełącznika `/e` wymusza kopiowanie wszystkich podkatalogów źródła, nawet jeżeli są puste. Katalogi nieistniejące na dysku przeznaczenia są tworzone.

Polecenie `Echo Kopiowanie zakończone...` wykonywane jest po skopiowaniu wszystkich wyspecyfikowanych plików. Powoduje ono wyświetlenie na ekranie informacji `Kopiowanie zakończone...`.

2. Zapisz program w pliku wsadowym *xcopy1.bat*.
3. Umieść w napędzie *A* dyskietkę.
4. Załóż w katalogu głównym dysku *C* folder *moje*.
5. W folderze *moje* utwórz folder *2005*.
6. W folderze *2005* utwórz plik *spis.txt*.
7. Uruchom program wsadowy.
8. Sprawdź, czy po zakończeniu działania programu na dyskietce pojawił się folder *moje* wraz z podfolderem *2005*. W podfolderze *2005* powinien znajdować się plik *spis.txt*.

Przykład 3.24.

Napisz program wsadowy, który będzie kopiował zawartość folderu *c:\moje* na dyskietkę umieszczoną w napędzie oznaczonym literą *A*. Program powinien umożliwiać skopiowanie plików **.doc* na jedną dyskietkę, a plików **.xls* na drugą.

1. Napisz program:

```
:: xcopy2.bat
:: Program kopiuje pliki *.doc z folderu c:\moje na jedną dyskietkę.
:: a pliki *.xls na inną.
@Echo off
cls
cd c:\moje
Echo Umieść dyskietkę w stacji A:
pause
xcopy *.doc A: /e
Echo Kopiowanie plików *.doc zakończone...
Echo Umieść dyskietkę w stacji A:
pause
xcopy *.xls A: /e
Echo Kopiowanie plików *.xls zakończone...
```

Program różni się od opisanego w przykładzie tym, że znajdują się w nim dwa polecenia kopiowania. Jedno służy do kopiowania plików **.doc*. Drugie umożliwia kopiowanie plików **.xls*.

Jeżeli po każdym wyświetleniu komunikatu Umieść dyskietkę w stacji A: włożysz do napędu inną dyskietkę, na pierwszą skopiowane zostaną arkusze kalkulacyjne, a na drugą dokumenty edytora *MS Word*.

2. Zapisz program w pliku wsadowym.
3. Umieść w napędzie *A* dyskietkę.
4. Załóż w katalogu głównym dysku *C* folder *moje*.
5. W folderze *moje* utwórz plik *misio.doc*.
6. W folderze *moje* utwórz folder *2006*.
7. W folderze *2006* utwórz plik *piesek.xls*.
8. Uruchom program wsadowy.
9. Sprawdź, czy po zakończeniu działania programu na dyskietce pojawił się folder *moje* wraz z podfolderem *2006*. W folderze *moje* powinien znajdować się plik *misio.doc*. W podfolderze *2006* powinien znajdować się plik *piesek.xls*.

Przykład 3.25.

Napisz program wsadowy, który będzie archiwizował zawartość folderu *c:\moje* na dyskietkę umieszczoną w napędzie oznaczonym literą *A*.

1. Napisz program:

```
:: xcopy3.bat
:: Program archiwizuje zawartość folderu c:\moje
```

```

:: na dyskietkę umieszczoną w napędzie a.
@Echo off
cls
cd c:\moje
Echo Umieść dyskietkę w stacji A:
pause
xcopy *.* A: /m /q /y
Echo Archiwizowanie plików zakończone...

```

Program różni się od opisanego w przykładzie 3.24 przełącznikami użytymi w wywołaniu polecenia `xcopy`. Takie ich połączenie powoduje, że:

- ♦ Kopiowane są pliki z ustawionym atrybutem archiwalnym. Po skopiowaniu atrybut jest wyłączany. Włączenie atrybutu powoduje np. wprowadzenie zmiany w pliku. Kopiowane są zatem tylko pliki, które od czasu wykonywania ostatniej kopii zostały zmienione.
- ♦ Podczas kopiowania nie są wyświetlane nazwy kopiowanych plików. Jest to wygodne, gdy kopiowanych jest wiele plików.
- ♦ Wyłączone jest monitowanie o potwierdzenie zastąpienia istniejącego pliku docelowego. Program nie zatrzymuje się w oczekiwaniu na pozwolenie zastąpienia starszej wersji pliku nowszą.

Przykład 3.26.

Napisz program, który umożliwi użytkownikowi określenie plików do skopiowania oraz lokalizacji i nazw skopiowanych plików. Kopiowane mają być również puste foldery, a także pliki z atrybutami.

Napisz program:

```

:: kopiuj.bat
:: Program oczekuje na podanie źródła i celu kopiowania.
:: Kopiowane są zarówno puste foldery jak i atrybuty plików.

@echo off
set /P co="Co skopiowac: "
set /P kopiuj="Gdzie skopiowac: "
xcopy /E /I /Q /H %co% %kopiuj%

```

Polecenie `set` z przełącznikiem `/p` ustawia wartość parametru jako wiersz danych wejściowych.

Zmienna (`co` lub `kopiuj`) określa zmienną, którą należy ustawić lub zmodyfikować.

Polecenie `xcopy` z przełącznikiem `/e` kopiuje wszystkie podkatalogi, nawet jeżeli są puste.

Przełącznik `/i` powoduje, że jeżeli parametr `co` jest katalogiem lub zawiera symbole wieloznaczne, a lokalizacja określona przez parametr `kopiuj` nie istnieje, polecenie `xcopy` przyjmuje, że w parametrze `kopiuj` podano nazwę katalogu, i tworzy nowy katalog. Następnie polecenie `xcopy` kopiuje wszystkie wskazane pliki do nowego katalogu.

Przełącznik /q sprawia, że polecenie xcopy pomija wyświetlanie komunikatów polecenia.

Przełącznik /h sprawia, że polecenie xcopy kopiuje pliki z atrybutami *ukryty* i *systemowy*.



Polecenie xcopy domyślnie nie kopiuje plików ukrytych lub systemowych.