

# Programowanie wsadowe

# Programowanie wsadowe

- Program wsadowy jest to ciąg poleceń trybu linii komend lub wywołań programów zapisany w pliku tekstowym o **rozszerzeniu .bat**

# Programowanie wsadowe

- Celem programowania wsadowego jest:
- przyspieszenie pracy
- personalizacja środowiska systemu operacyjnego

# Programowanie wsadowe

- W języku programowania wsadowego można stosować wszystkie komendy dostępne w trybie linii komend.

# Do sterowania działaniem programu wsadowego służą komendy sterujące:

- **call** - uruchamia programu wsadowego z poziomu innego programu wsadowego, a następnie powrót do programu źródłowego
- **echo** - wyświetla komunikaty na ekranie, oraz może służyć do włączania lub wyłączania wyświetlania komunikatów

# Do sterowania działaniem programu wsadowego służą komendy sterujące:

- **for** - powtarza komendę dla zestawu zmiennych
- **goto** - przechodzi do innej części programu
- **If** - wykonuje komendy w zależności od spełnienia warunku

# Do sterowania działaniem programu wsadowego służą komendy sterujące:

- **pause** - zatrzymuje wykonanie programu do czasu wciśnięcia dowolnego klawisza
- **rem** - komentarz
- **shift** - przesuwa parametry programu
- **choice**- pozwala użytkownikowi wybrać opcję podczas wykonywania programu

# HELP

- Informacje o sposobie wywołania komendy uzyskuje się poprzez wpisanie nazwy komendy z parametrem `/?`
- np.: `if /?`



# Uruchamianie plików .bat

- Programy wsadowe uruchamia się podając w trybie **konsolowym nazwę pliku wsadowego** (najlepiej łącznie z rozszerzeniem).
- Plik wsadowy można także uruchomić w trybie okienkowym wybierając go poprzez **kliknięcie klawiszem myszki** - tak jak każdy inny program.

# Zatrzymać wykonanie programu

- **Zatrzymać wykonanie** programu wsadowego można poprzez naciśnięcie kombinacji klawiszy **Ctrl+S** lub **Pause**.
- Wykonanie programu **jest kontynuowane** po naciśnięciu dowolnego klawisza.

# Zatrzymać wykonanie programu

- **Przerwać działanie** programu wsadowego można poprzez naciśnięcie kombinacji klawiszy **Ctrl+C** lub **Ctrl+Break**.

# Używanie Notatnika do tworzenia skryptów.

- Skrypty są plikami tekstowymi, więc do ich tworzenia można używać programu **Notatnik**.
- Należy jednak uważać na ustawienia domyślne np. zapisując pod nazwę **skrypt.bat** program zapisze **skrypt.bat.txt**, czyli **NIE jest to skrypt**

# Wyświetlanie komunikatów

- Komunikaty w programach wsadowych można wyświetlać korzystając z instrukcji echo. Składnia tego polecenia:

```
echo <Treść komunikatu>
```

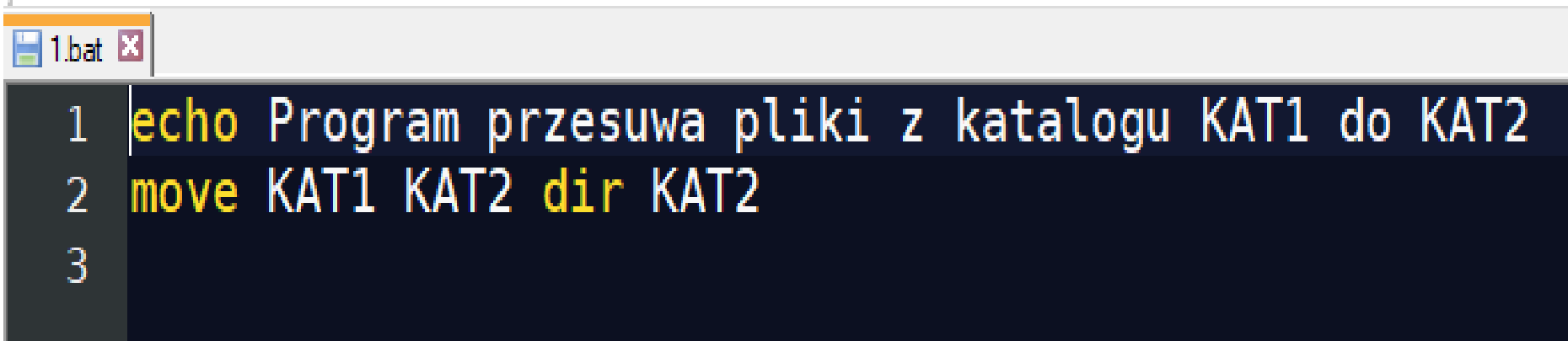
# Wyświetlanie komunikatów

- Aby wyświetlić pustą linię należy użyć polecenia echo. (z kropką na końcu, bez spacji).

echo.

- (z kropką na końcu, bez spacji).

# Przykład

A screenshot of a Windows command prompt window. The title bar shows a file icon, the name '1.bat', and a close button. The command prompt has a dark background with white text. The text is as follows:

```
1 echo Program przesuwa pliki z katalogu KAT1 do KAT2
2 move KAT1 KAT2 dir KAT2
3
```

Skrót do folderu pliki wsadowe

# Wyłączyć wyświetlanie instrukcji

- Komenda echo może być użyta do ograniczenia wyświetlania wykonywanych instrukcji programu.
- I tak, aby wyłączyć wyświetlanie wykonywanych instrukcji należy użyć instrukcji:

```
echo off
```



# Włączyć wyświetlanie instrukcji

- Aby włączyć wyświetlanie wykonywanych instrukcji:

```
echo on
```

# znak @

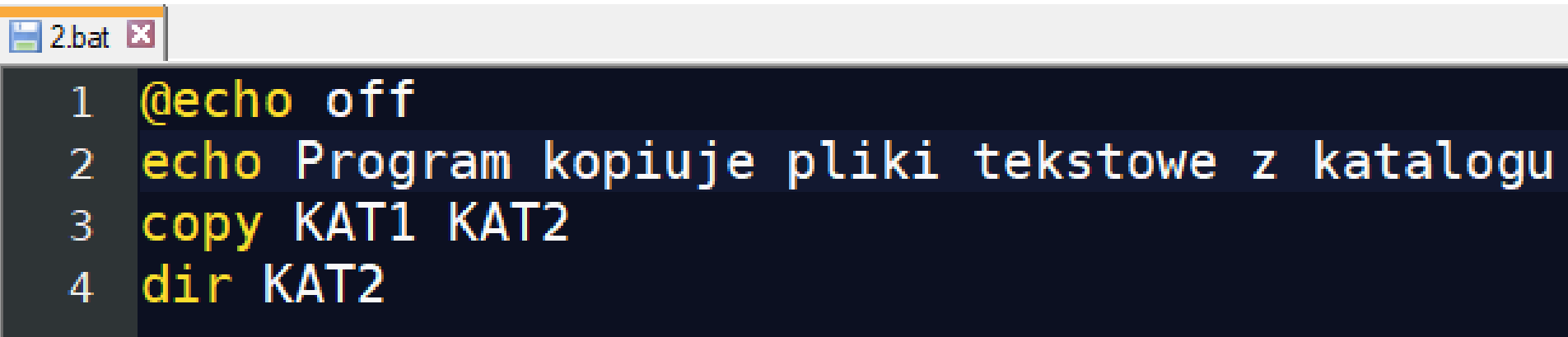
- Efekt podobny do użycia echo off dla pojedynczej instrukcji można osiągnąć poprzedzając instrukcję znakiem @.

# @echo off

- Zazwyczaj nie jest pożądane wyświetlanie poszczególnych poleceń ze skryptu, dlatego też większość skryptów będzie się zaczynać od tego polecenia.

```
@echo off.
```

# Przykład



```
2.bat  
1 @echo off  
2 echo Program kopiuje pliki tekstowe z katalogu  
3 copy KAT1 KAT2  
4 dir KAT2
```

Skrót do folderu pliki wsadowe

# Komenda pause

- Zatrzymanie programu można wymusić korzystając z komendy *pause*. Instrukcja *pause* wyświetla komunikat :

Naciśnij dowolny klawisz aby kontynuować...

- a następnie oczekuje na wciśnięcie dowolnego klawisza.

# Przykład

```
2.bat x
1 @echo off
2 echo Program kasuje pliki tekstowe w KAT1
3 echo Zaczynam kasować...
4 pause
5 del KAT1\*.txt
6 cls
7 dir KAT1
8
```

# Ćwiczenia do wykonania

1. W katalogu, do którego masz prawo do zapisu, **utwórz skrypt o nazwie a.bat**
2. **Skrypt ma utworzyć następującą strukturę katalogów: . kat1 kat2 podkatalog jeden**  
(proszę zachować spację w nazwie podkatalogu)

```
├─ kat1
├─ kat2
└─ podkatalog  jeden
```

# Ćwiczenia do wykonania

3. W podkatalogu proszę **utworzyć plik nazwa.txt**, który będzie zawierał ścieżkę do tego katalogu (należy wykorzystać polecenie `cd` i przekierowanie strumienia)



# Odpowiedź do ćwiczenia

2.bat

```
1 @echo off
2 MD .\kat1
3 MD .\kat2\ "podkatalog jeden "
4 cd .\kat2\ "podkatalog jeden "
5 cd > nazwa.txt
```

# Etykieta i skok do etykiety goto

- Etykieta jest to nazwa danego wiersza w skrypcie. Etykiety definiuje się pisząc nazwę etykiety poprzedzoną dwukropkiem np.:

```
:etykieta
```

# Etykieta i skok do etykiety goto

- Poprzez etykiety i instrukcje skoku ***goto*** można zmienić kolejność wykonywania instrukcji programu.
- Składnia polecenia skoku wygląda następująco:

```
goto etykieta
```

# Przykład a.bat

```
1 @echo off
2 :start
3 echo Plik wsadowy
4 goto start
```

[Skrót do folderu pliki wsadowe](#)

# Przyjmowanie danych

- Zaczniemy od przyjmowania danych w oknie programu. Nazwy zmiennych podajemy ograniczone procentami.

```
@echo off  
echo Jak masz na imię?  
set /p imie=  
echo Witaj %imie%  
pause
```

# Przyjmowanie danych - parametry

- Programy wsadowe mogą być wywoływane z parametrami
- **np. dla polecenia copy plik1 plik2**  
pierwszym argumentem jest plik1, a drugim plik2.

# Przyjmowanie danych - parametry

- Dostęp do parametrów z poziomu programu wsadowego odbywa się przez użycie symboli **%0 do %9**.
- Symbol **%0** oznacza nazwę wykonywanego programu wsadowego.
- Symbole od **%1 do %9** odpowiadają kolejnym parametrom.

# Przykład

w1.bat

```
1 @echo off
2 echo Program kopiuje pliki tekstowe
3 pause
4 copy %1\*.txt %2
5 cls
6 dir %2 /p
```

```
rs\renata_2>Skrypt.bat Ala Ola
```



# CALL

- **CALL** - wywołuje inny plik wsadowy, a po jego zakończeniu powraca do macierzystego. Program może też wywoływać samego siebie.
- **Składnia:**  
call [dysk:][ścieżka]nazwa pliku[parametry wsadowe]

# Przykład pliku z komendą call

```
1 @ECHO OFF
2 :start
3 IF "%1"==" " GOTO koniec
4 CALL del.bat %1
5 SHIFT
6 GOTO start
7 :koniec
```

# Instrukcja warunkowa if

- W programach wsadowych można używać instrukcji warunkowej **if**, której ogólna zasada sprowadza się do sprawdzenia określonego warunku logicznego i przy jego spełnieniu wykonania dowolnej instrukcji.

# Instrukcja warunkowa if

```
1 IF [NOT] ERRORLEVEL liczba polecenie  
2  
3 IF [NOT] "napis1"=="napis2" polecenie  
4  
5 IF [NOT] EXIST plik1 polecenie
```

**Program sprawdza istnienie katalogu i w zależności czy istnieje usuwa go bądź tworzy.**

```
b.bat x
1 if exist c:\katalog\nul (rd c:\katalog)
2 else (md c:\katalog)
```

[Skrót do folderu pliki wsadowe](#)

```
1 @ECHO OFF
2 IF EXIST lista.txt GOTO Usuwaj
3 ECHO Nie ma takiego pliku...
4 PAUSE
5 GOTO Koniec
6
7 :Usuwaj
8 ECHO Usuwanie pliku lista.txt...
9 DEL lista.txt > NUL
10 ECHO Wykonane...
11 PAUSE
12
13 :Koniec
```

[uruchom program 3.bat](#)

# Wykorzystanie instrukcji warunkowych w celu znalezienia pliku.

```
if exist c:\moje\zm.txt goto 1
if not exist c:\moje\zm.txt goto 2
:1
echo Znaleziono plik.
pause >nul
exit
:2
echo Nie znaleziono pliku.
pause >nul
exit
```

# PLIK BAT Z WYKORZYSTANIEM PARAMETRÓW

```
1 @ECHO OFF
2 IF "%1"==" " GOTO error
3 REM Jeżeli nie podano parametru to podaje składnię.
4 REM Jeżeli podano parametr to wypisze zdanie.
5 ECHO Podano liczbę %1.
6 GOTO koniec
7 :error
8 ECHO Prawidłowa składnia: %0 liczba
9 :koniec
```

uruchom program przyklad40.bat



# Komunikacja z użytkownikiem

- W programowaniu wsadowym komunikacja z użytkownikiem jest zagadnieniem skomplikowanym.
- Program **choice.com** wyświetla komunikat, zgłasza oczekiwanie na odpowiedź, wczytuje znak z klawiatury, a następnie ustawia kod wyjścia, który służy do identyfikacji wciśniętego klawisza.

## choice *tekst*

- Powoduje wyświetlenie na ekranie komunikatu w postaci: *tekst [Y,N]?*  
Jeśli użytkownik wciśnie **Y** lub **y** to program choice ustawi wartość kodu wyjścia na 1, jeśli wciśnie **N** lub **n** kod wyjścia przyjmie wartość 2.

# choice *tekst*

- Wciśnięcie każdego innego klawisza powoduje wygenerowanie krótkiego dźwięku, a program choice będzie czekał na odpowiedź.

# choice /c:lista

- Przy pomocy opcji **/c:lista** możemy rozszerzyć zakres dopuszczalnych odpowiedzi.
- Opcja **/c** umożliwia ustawienie rozróżniania **dużych i małych** liter.

- **Przykładem wykorzystania choice może być system menu, z którego wybieramy opcje.**

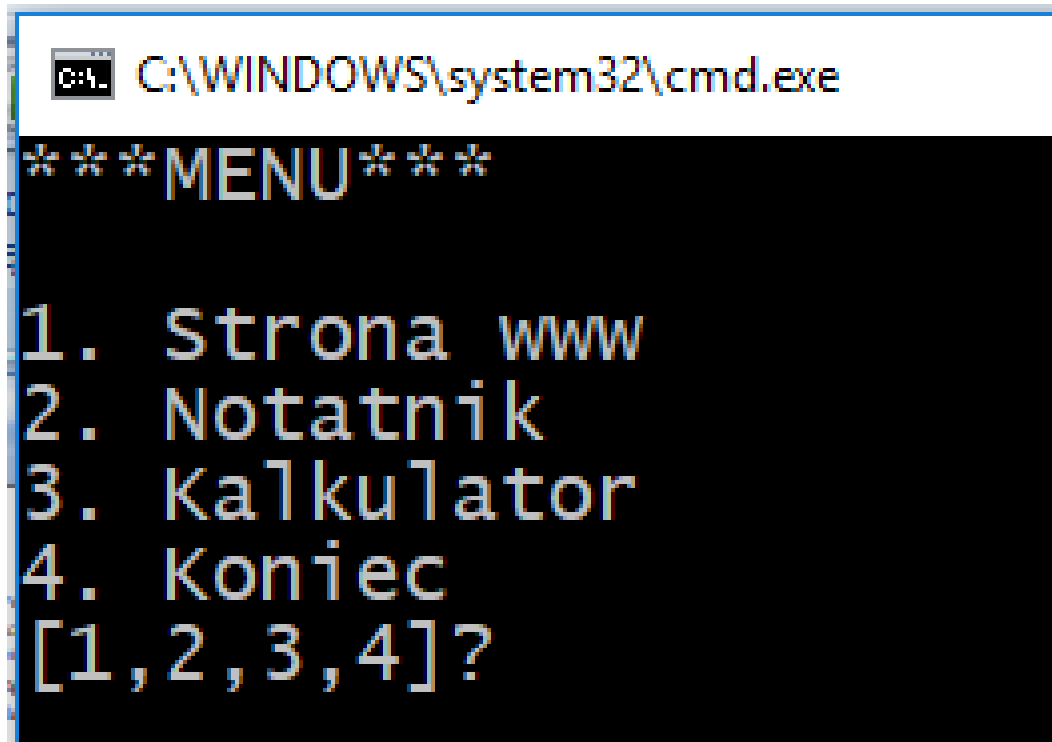
C:\WINDOWS\system32\cmd.exe

```
czy masz 18 lat?  
1 - pełnoletni  
2 - niepełnoletni  
[1,2]?
```

```
1 @echo off
2 echo czy masz 18 lat?
3 echo 1 - pelnoletni
4 echo 2 - niepelnoletni
5 choice/c:12
6 if errorlevel = 2 goto mlody
7 if errorlevel = 1 goto stary
8
9 :stary
10 echo wybrales 1 czyli masz wiecej niz 18 lat!
11 pause
12 goto koniec
13
14 :mlody
15 echo wybrales 2 czyli nie masz 18 lat!
16 pause
17 goto koniec
18
19 :koniec
```

[Uruchom program 4.bat](#)

- **Menu wyboru:**



```
C:\WINDOWS\system32\cmd.exe
***MENU***
1. Strona www
2. Notatnik
3. Kalkulator
4. Koniec
[1,2,3,4]?
```

```
1 @echo off
2     :BEG
3     cls
4     echo ***MENU***
5     echo.
6     echo 1. Strona www
7     echo 2. Notatnik
8     echo 3. Kalkulator
9     echo 4. Koniec
10    choice /c:1234
11    if errorlevel 4 goto END
12    if errorlevel 3 goto CALC
13    if errorlevel 2 goto NOTE
14    if errorlevel 1 goto WWW
15    goto BEG
16    :CALC
17    start calc
18    goto BEG
19    :NOTE
20    start notepad
21    goto BEG
22    :WWW
23    start http://banasik.pisz.pl
24    goto BEG
25    :END
```

[Uruchom  
przykład](#)



```
1 @ECHO OFF
2 CHOICE /C:1234 /d:2 /T:10
3 IF ERRORLEVEL 4 GOTO dorota
4 IF ERRORLEVEL 3 GOTO celina
5 IF ERRORLEVEL 2 GOTO beata
6 IF ERRORLEVEL 1 GOTO anna
7 :anna
8 ECHO Wybrałeś Anna
9 GOTO koniec
10 :beata
11 ECHO Wybrałeś Beata
12 GOTO koniec
13 :celina
14 ECHO Wybrałeś Celina
15 GOTO koniec
16 :dorota
17 ECHO Wybrałeś Dorota
18 :koniec
```

[Uruchom  
przykład](#)

# ZADANIE

C:\WINDOWS\system32\cmd.exe

```
chcesz sie przywitac czy pozegnac?  
Press any key to continue . . .  
1 - powitanie  
2 - pozegnanie  
[1,2]?1  
wybrales 1, witam!  
Press any key to continue . . .
```

```
1 @echo off
2 echo chcesz sie przywitac czy pozegnac?
3 pause
4 echo 1 - powitanie
5 echo 2 - pozegnanie
6 choice/c:12
7 if errorlevel = 2 goto papa
8 if errorlevel = 1 goto czesc
9 :czesc
10 echo wybrales 1, witam!
11 pause
12 goto koniec
13 :papa
14 echo wybrales 2, zegnam!
15 pause
16 goto koniec
17 :koniec
```

# Instrukcja iteracyjna FOR

- **Instrukcje iteracyjne** (powtarzające, zwane też pętlami) służą w językach programowania do wielokrotnego wykonywania podobnych czynności. Ogólnie dzieli się instrukcje iteracyjne na pętle o znanej lub niemożliwej do przewidzenia z góry liczbie powtórzeń.

# Instrukcja iteracyjna FOR

- Instrukcja iteracyjna **FOR** pliku wsadowego jest pętlą o znanej liczbie powtórzeń i następującej składni:

```
FOR %%X IN (lista) DO komenda
```

```
FOR %%X IN (lista) DO komenda
```

- Argument **lista** jest obowiązkowy i zawiera pooddzielane znakami odstępu nazwy.
- Po znakach **'%%'** podajemy tzw. zmienną która będzie kolejno przybierać wszystkie możliwe wartości spośród wartości podanych na liście (**lista**).
- Dla każdej możliwej wartości **zmiennej X** zostanie wykonane polecenie **komenda**.

# Przykład prezentujący iteracje

```
1 @echo off
2 cls
3 FOR %%W IN (pomidor, rzodkiewka, ziemniak) DO echo To jest %%W
4 pause
```

[Uruchom przykład](#)

# Przykład

Program wypisuje liczby od 1 do 100

```
1 @echo off
2 for /L %%i in (1,1,100) do echo %%i
3 pause
```

[Uruchom przykład](#)



```
FOR %I IN (C:\*.* ) DO @echo %I
```

W przypadku, gdy komenda for jest wpisywana w linii poleceń, to nazwa zmiennej musi być poprzedzona jednym znakiem %

```
FOR %%I IN (C:\*.* ) DO echo %%I
```

W przypadku, gdy komenda ta znajduje się w pliku wsadowym, wymagane jest poprzedzenie jej %%

# Wykonanie polecenia TYPE dla plików z rozszerzeniem .DOC i .BAT

```
1 @echo off
2 cls
3   FOR %%M IN (*.doc,*.bat) DO TYPE %%M
4 pause
```

```
FOR /D %I IN (C:\*.* ) DO @echo %I
```

- Przełącznik /D powoduje, że zamiast plików wyszukiwane są katalogi.

# Przykład zastosowania pętli w pętli z podanymi wartościami:

```
1 @echo off
2 cls
3 FOR %%o IN (garnek, patelnia) DO ( FOR %%w IN (pomidorem, rzodkiewka) DO echo To jest %%o z %%w )
4 pause
```

**@echo off**

[Uruchom przykład](#)

**FOR %%o IN (garnek, patelnia)**

**DO ( FOR %%w IN (pomidorem, rzodkiewka)**

**DO echo To jest %%o z %%w )**

# Przykład pętli o niemożliwej do ustalenia z góry liczbie powtórzeń:

```
1  :Start
2  CHOICE /C:TN "Powtórzyć ?"
3  IF ERRORLEVEL 2 GOTO Koniec
4  GOTO Start
5
6  :Koniec
```