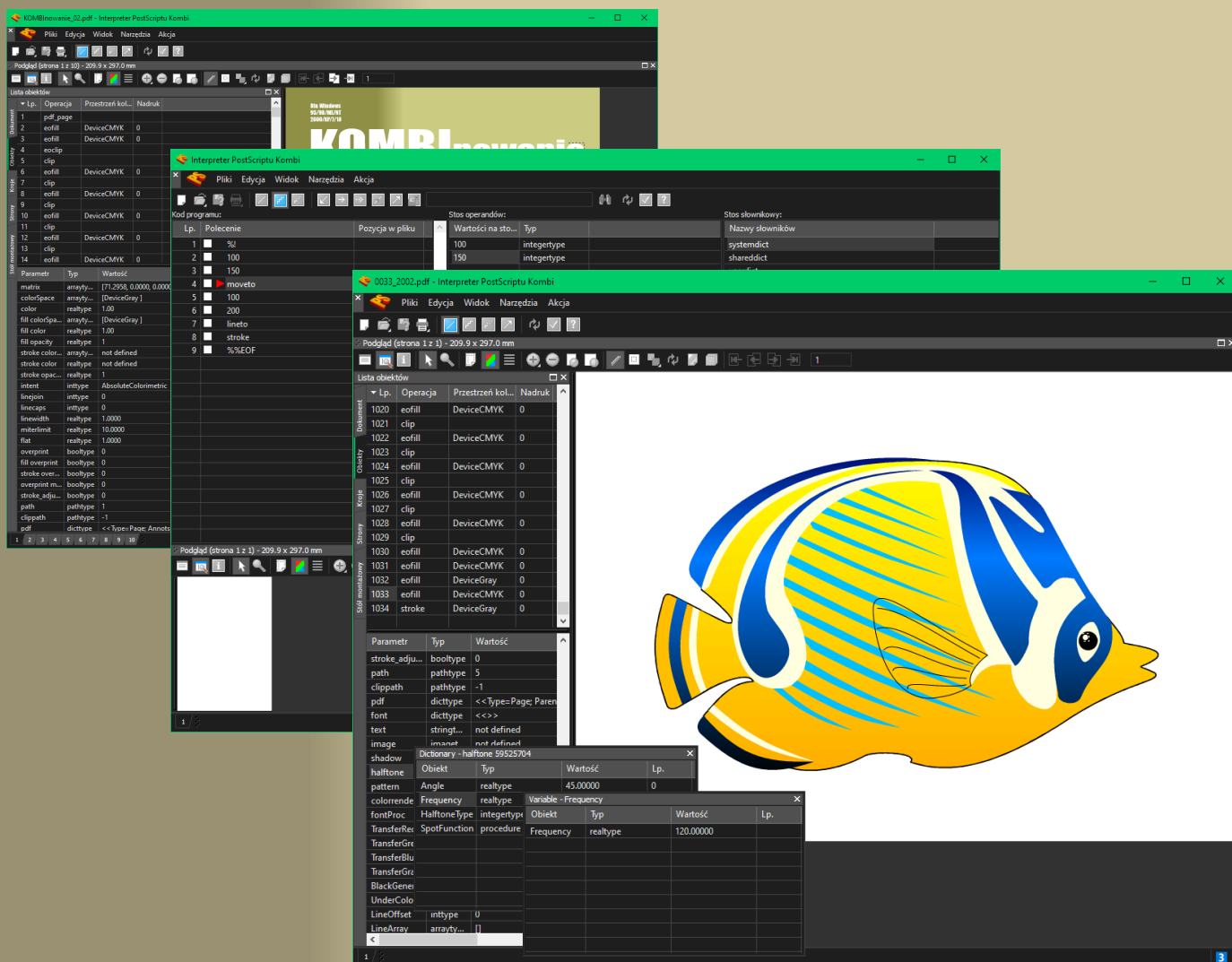


Dla Windows  
95/98/ME/NT  
2000/XP/7/10

# KOMBI nowanie

Poradnik użytkownika programu KOMBI

Numer 2, grudzień 2020 • Gazeta wychodzi kiedy chce



**W numerze:**

- Interpreter PostScriptu
- Co i jak? klatka z filmu

**System komputerowego składu dokumentów ze słownikiem ortograficznym i edytorami grafiki wektorowej i rastrowej**

75-814 Koszalin, ul. Zakole 20/7, ps.3n.com.pl, e-mail: office@3n.com.pl

# Interpreter PostScriptu

## Cały bieżący numer biuletynu poświęcamy jednemu rozszerzeniu pakietu Kombi, tj. interpreterowi PostScriptu

Wersji ósmej pakietu Kombi pojawiło się nowe rozszerzenie programu, tj. Interpreter PostScriptu.

W wersji **dziewiątej** Interpreter został znacznie rozbudowany i jego funkcjonalność jest teraz o wiele większa niż to opisano w niniejszej publikacji, którą dedykujemy przede wszystkim osobom, które chcą wykorzystać interpreter do programowania w języku PostScript. Pełna i aktualna dokumentacja znajduje się na stronie domowej produktu, tj. [ps.3n.com.pl](http://ps.3n.com.pl).

Aby wyjaśnić, czym jest interpreter odpowiemy sobie najpierw na pytanie:

### Czym jest PostScript?

PostScript jest **językiem programowania** przeznaczonym do opisywania zawartości strony. Z faktu, że jest to język programowania wynika podstawowa różnica między nim, a innymi formatami stosowanymi w DTP. Otóż, jeśli mamy do czynienia z dowolnym innym formatem – dane muszą być w tym formacie zapisane wprost (tzn. nie ma w pliku zmienionych, ale wyłącznie stałych; wszystkie dane są wyliczone w programie zapisującym i w pliku zapisane są wyniki tych obliczeń). W przypadku PostScriptu – ponieważ jest to język programowania – w pliku mogą występować zmienne, których wartość w momencie zapisywania pliku nie jest ustalona.

Zalóżmy np., że chcemy wydrukować katalog krojów. W przypadku formatów innych niż PostScript, aby wykonać to zadanie musimy wcześniej założyć na jakim formacie ten katalog będzie drukowany. I z tego założenia oraz przyjętej wielkości znaków wyniknie nam określona liczba wierszy tekstu, którą możemy pomieścić na założonym arkuszu. Jeśli tak przygotowany plik wydrukujemy na innym formacie niż założyliśmy, to liczba wydrukowanych wierszy będzie zawsze taka sama, zmieniać się będzie co najwyżej wielkość znaków. W przypadku PostScriptu – można tak przygotować plik, żeby wielkość arkusza nie była ustalona z góry, ale żeby była odczytywana „w locie”. W takim rozwiązaniu, przy założonej wielkości znaków liczba wydrukowanych wierszy na stronie będzie zależna od wybranego formatu.

Innym przykładem może być wykorzystywanie różnego rodzaju nowinek technicznych dostępnych w drukarce. W pliku postscriptowym mogą znajdować się zapytania sprawdzające, czy drukarka obsługuje jakąś specyficzną funkcję i w zależności od wyniku tego zapytania mogą być drukowane różne dane (np. bitmapy w różnych przestrzeniach kolorów).

W związku z tym, aby taki plik otworzyć i „zrozumieć”, należy dokonać analizy zapisanych w nim zmiennych i przeprowadzić na nich obliczenia tak, by ustalić ich właściwe wartości. Programem, który taką analizę wykonuje jest interpreter PostScriptu.

Postawmy sobie wobec tego kolejne pytanie:

### W jaki sposób zapisane są dane w pliku postscriptowym?

Plik postscriptowy jest plikiem tekstowo-binarnym. To znaczy, że jeśli otworzymy go np. w Notatniku, to część danych będziemy mogli przeczytać (część tekstową), natomiast fragmenty binarne będziemy widzieć w postaci „krzaczków”.

Jak to się dzieje, że interpreter potrafi te krzaczki zrozumieć?

Koncepcja PostScriptu oparta jest na argumentach i operatorach zapisywanych w notacji ONP (odwrotna notacja polska). Oznacza to, że w zapisie występują najpierw parametry, a potem nazwa operatora. Np. zapis **10 20 add** oznacza, że należy wziąć dwie liczby (**10** i **20**), a następnie je dodać (operator **add** oznacza dodawanie). Jeśli w pliku są dane binarne, to są one poprzedzone odpowiednim operatorem (np. operatorem **image**), przed którym stoją argumenty, z których wynika ile bajtów danych za operatorem stanowi dane binarne. Tak więc interpreter analizuje dane krok po kroku (bajt po bajcie) i buduje z nich tzw. **tokeny**. Każdy uformowany token jest porównywany z listą operatorów i jeśli token zostanie znaleziony na tej liście – to interpreter od tego momentu wie, jaki operator został odczytany z pliku. Jeśli token nie zostanie znaleziony na liście operatorów, to albo jest zmienną, albo liczbą. Jeśli jest zmienną – to program porównuje ten token z listą zdefiniowanych już zmiennych i jeśli ją znajdzie – to odczytuje wartość przypisaną tej zmiennej, a jeśli nie znajdzie – to interpreter zgłasza nam błąd. W każdym z przypadków – po przeanalizowaniu tokenu wiemy, że albo odpowiada mu jakaś wartość, albo reprezentuje on operator.

Jeśli token reprezentuje wartość – to wartość ta jest przekazywana na tzw. stos. Stos – to taka struktura, jak stos kartek na biurku. Możemy położyć kolejną kartkę na stos i zawsze możemy wziąć jedną kartkę ze stosu. Zalóżmy, że w pliku postscriptowym znajduje się ciąg tokenów: **10 20 add**. Interpreter formuje pierwszy token (**10**) i stwierdza, że nie jest to operator, tylko liczba. Przekazuje więc tę wartość na stos, który wygląda od tego momentu tak:

— początek stosu  
**10**

Po pobraniu drugiego tokenu – stos wygląda tak:

— początek stosu  
**10**  
**20**

Token trzeci zostaje zidentyfikowany jako operator **add**. Operator ten „konsumuje” ze stosu dwie dane, a wynik odkłada na stos. Po wykonaniu operatora **add** stos będzie wyglądał tak:

— początek stosu  
**30**

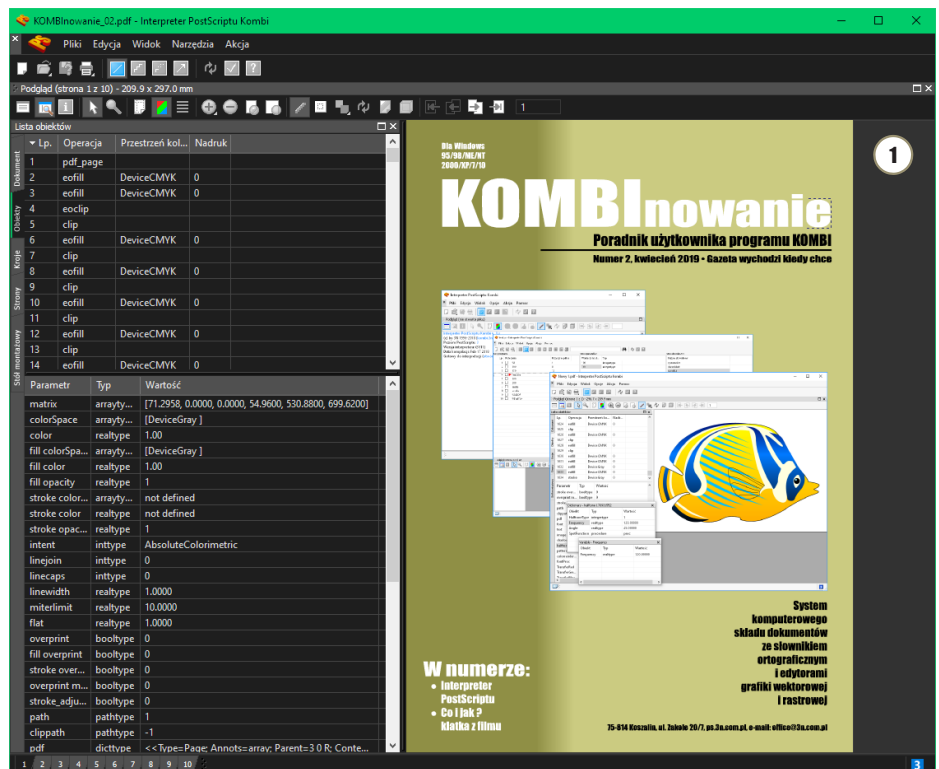
W dokumentacji języka PostScript sytuację tę opisuje się tak: **num1 num2 add num3** co oznacza, że operator **add** pobiera dwie liczby i zwraca jedną.

Zapis: **num1 num2 moveto** – oznacza natomiast, że operator **moveto** pobiera dwie liczby, ale nie zwraca nic. I odwrotnie: **— currentcmykcolor num1 num2 num3 num4** oznacza, że operator **currentcmykcolor** nie pobiera danych i zwraca na stos cztery liczby.


Jak w każdym języku programowania, również i tu można definiować zmienne, np. zapis: **/test 2 def** oznacza, że zmiennej **test** przypisaliśmy wartość **2**. Nazwy zmiennych poprzedzone są znakiem **/**. Zmienne przechowywane są w tzw. słownikach. Słownik, to zestaw par klucz → wartość. Interpreter przechowuje słowniki na stosie (lista **Stos słownikowy**). Kiedy program szuka wartości dla danego klucza przegląda stos słowników zaczynając od słownika znajdującego się na jego szczycie. Jeśli w słowniku tym nie zostanie znaleziona dana zmienna – program przeszukuje następny słownik, itd. aż do znalezienia zmiennej (lub stwierdzenia, że zmienna nie istnieje, co oczywiście spowoduje przerwanie pracy interpretera z podaniem odpowiedniego komunikatu błędu). Słownik o nazwie **systemdict** zawiera predefiniowane zmienne i operatory.

Po tym wstępnym omówieniu podstaw PostScriptu spójrzmy na okno interpretera i popatrzmy dalej, jak on działa.

Prezentowany na rys. 1 może pracować w trzech trybach.





## Tryb krokowy



Aby zobaczyć jak pracuje interpreter możemy użyć trybu krokowego (ikona  lub polecenie **Tryb krokowy z prekompilacją** z menu **Widok**). W trybie tym plik jest najpierw wczytywany do pamięci i dzielony na tokeny i dane. Założmy, że interpretowany będzie następujący plik:

```
%!
100 150 moveto
100 200 lineto
stroke
%%EOF
```

Po otwarciu tego pliku w trybie krokowym odczytane z pliku tokeny zostaną wyświetlone na liście **Kod programu** (rys. 2). Zawartość tej listy jest tworzona z tokenów pobranych z interpretowanego pliku, ale można też ręcznie zmieniać (także dodawać lub usuwać) wpisy na tej liście (**F2** – edytuje, **Insert** – wstawia pozycję, **Delete** – usuwa pozycję).

Poniżej (pod listą **Kod programu**) widzimy okno komunikatów interpretera, w którym będą wyświetlane informacje o pracy programu, a także – wyświetlany będzie obraz graficzny zinterpretowanego pliku. Między oknem komunikatów, a oknem podglądu przełączamy się ikonami  i .

Po prawej stronie widzimy szereg list, których zastosowanie omówię w dalszej części artykułu.

Ikona  (lub klawiszem **Enter**) powodujemy interpretację kolejnych wpisów na liście **Kod programu**. Pozycja, która będzie wykonana w następnym kroku jest oznaczona na liście ikoną  (można przesunąć ten wskaźnik używając klawiszy kursora łącznie z klawiszem **Ctrl**).

W naszym przykładzie pierwszy wpis rozpoczyna się znakiem **%**. Znak ten oznacza komentarz, tzn. że cały token poprzedzony tym znakiem jest pomijany. Tak więc po wykonaniu pierwszego kroku stan interpretera nie ulega zmianie (token pierwszy jest komentarzem).

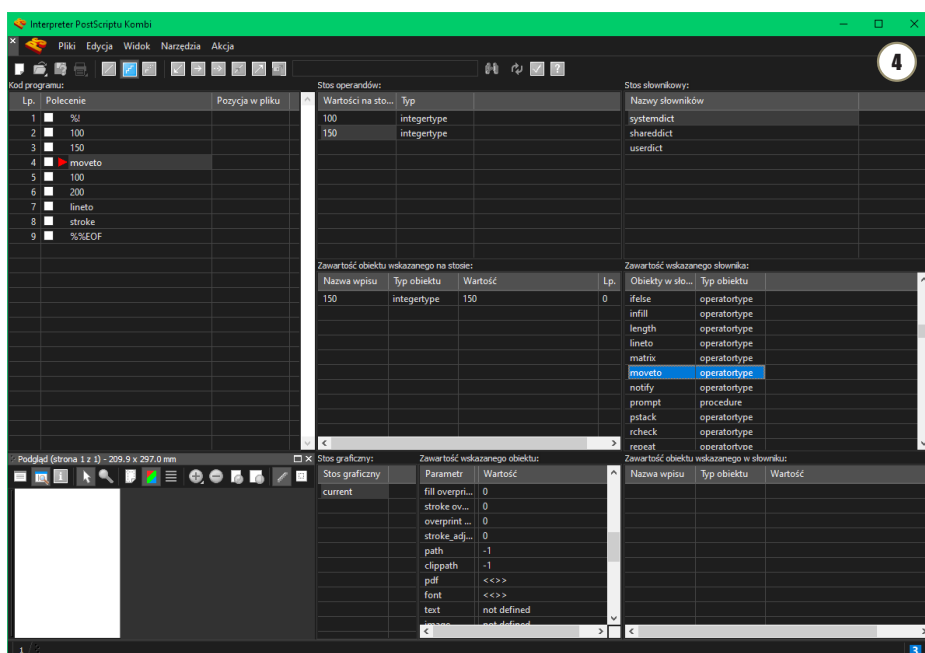
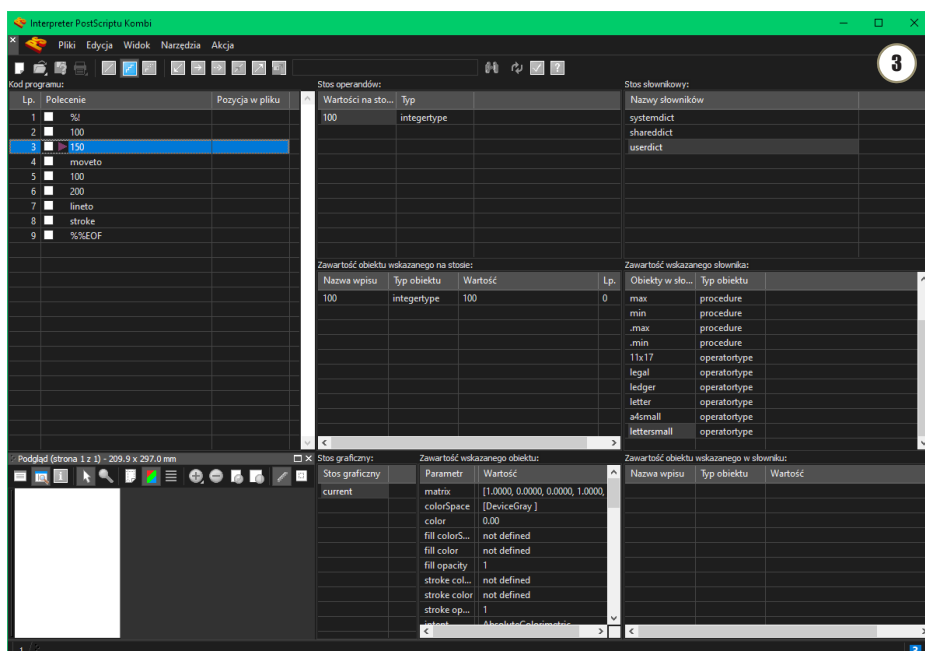
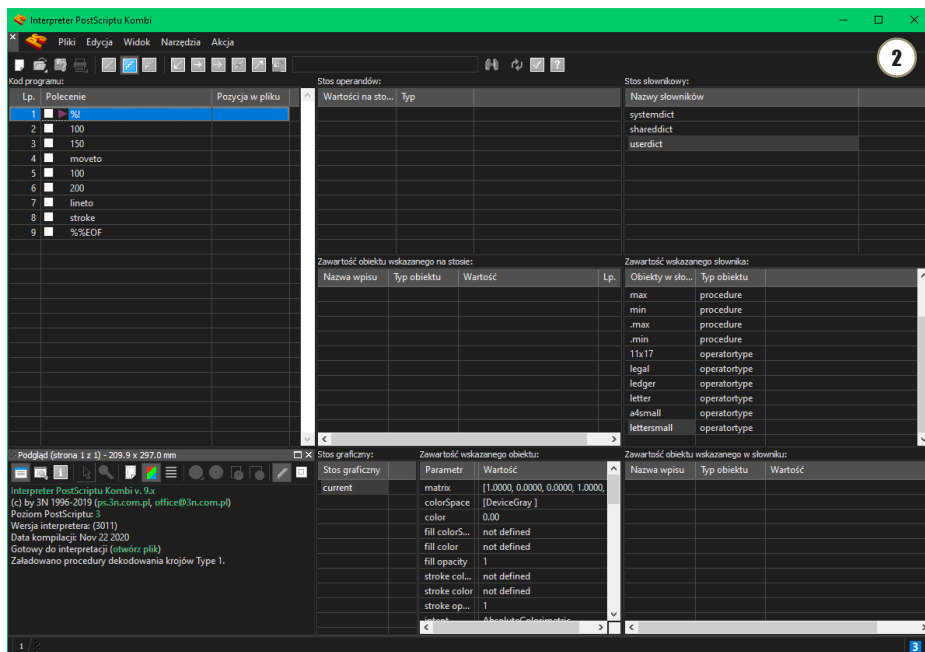
Krok 2. W czasie wykonywania tokena nr 2 interpreter stwierdza, że jest to liczba. W związku z tym, liczba ta jest przenoszona na **Stos operandów** (rys. 3). Jeśli liczba jest współzrędną, to jest ona wyrażona w punktach typograficznych. Jeden punkt typograficzny jest równy 1/72 cala.

Krok 3. Token nr 3 również jest liczbą, więc liczba ta przenoszona jest na stos operandów (rys. 4).

Krok 4. W czasie wykonywania kroku czwartego interpreter stwierdza, że token nie jest liczbą, próbuje więc odszukać go na liście tokenów. Lista ta znajduje się w słowniku **systemdict** (lista **Stos słownikowy**). Możemy wybrać pozycję **systemdict** na liście **Stos słownikowy**, co spowoduje pokazanie na liście **Zawartość wskazanego słownika** wpisów znajdujących się w wybranym słowniku.

Dla słownika **systemdict** widzimy, że znajduje się na niej operator **moveto** (rys. 4). Operator ten pobiera ze stosu dwie dane (są to współrzędne x i y). Co się dzieje z tymi współzrędnymi? Otóż są one dodawane do tzw. ścieżki (ang. path). Ścieżka jest elementem obiektu o nazwie **stan grafiki** (ang. graphic state). Stan grafiki poza ścieżką przechowuje wszystkie atrybuty związane z rysowanymi obiektami (kolory, grubości linii, kroje, itp.).

Stany grafiki są przechowywane na liście nazywanej w programie **Stos graficzny** (rys. 4). Przed wykonaniem kroku 4 pozycja **path** na tej liście zawiera wartość **-1**, co oznacza, że ścieżka nie jest jeszcze zdefiniowana (rys. 4). Po wykonaniu kroku 4, ze stosu operandów znikają dwie liczby, a pozycja **path** (na liście **Stos graficzny**) przyjmuje wartość **1**, co oznacza, że operator **moveto** pobrał dane ze stosu operandów i wstawił do ścieżki jeden punkt



(o współrzędnych odpowiadających wartościom pobranym ze stosu) – rys. 5.

Tokeny na pozycjach 5 i 6 odpowiadają tokenom 2 i 3. Z kolei token 7 – odpowiada tokenowi 4, z tym że do ścieżki dodawana jest informacja o tym, że poprzedni punkt z bieżącym należy połączyć linią (**lineto**). Po wykonaniu kroku 7 pozycja **path** na liście **Stos graficzny** zawiera wartość 2 (ścieżka zawiera dwa punkty) – rys. 6.

Krok 8. Pozycja ósma na liście **Kod programu** zawiera token **stroke**. Token ten jest operatorem i powoduje narysowanie linii wzdłuż punktów zawartych w ścieżce. Na rys. 7 widzimy, że po wykonaniu kroku ósmego w okienku podglądu pojawiła się pionowa linia. Linia ta jest zbudowana na ścieżce, którą wcześniej utworzyły operatory **moveto** i **lineto**. Jednocześnie widać, że **path** na liście **Stos graficzny** znowu ma wartość -1 (ścieżka niezdefiniowana), a więc operator **stroke** „skonsumował” ścieżkę. W tym stanie interpreter jest gotowy do analizy następnych danych. W naszym przykładzie jest tu koniec danych (**%%EOF**).


Możemy jeszcze zadać pytanie jakie parametry zostały użyte do narysowania linii?

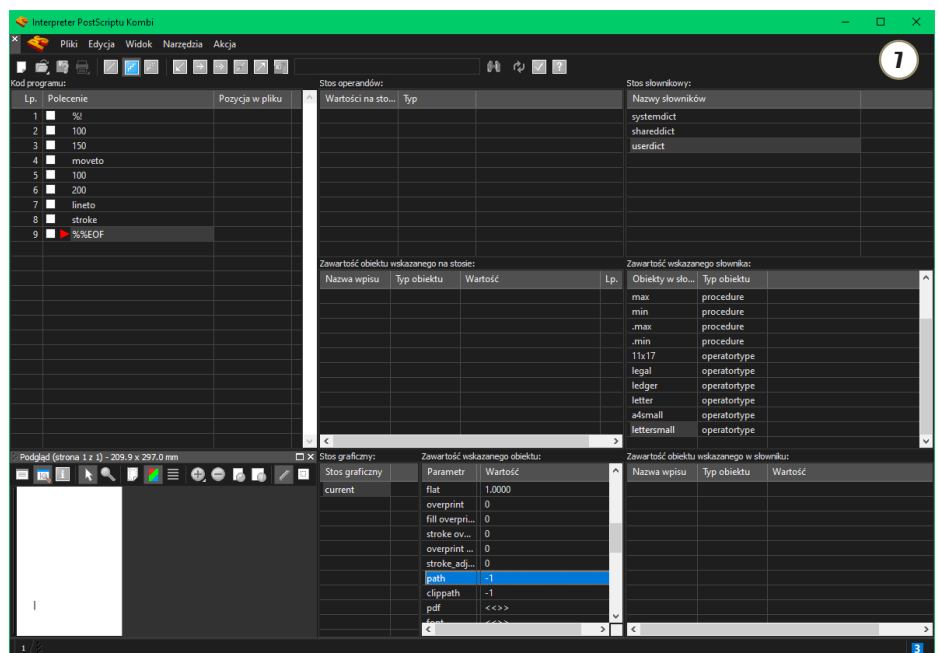
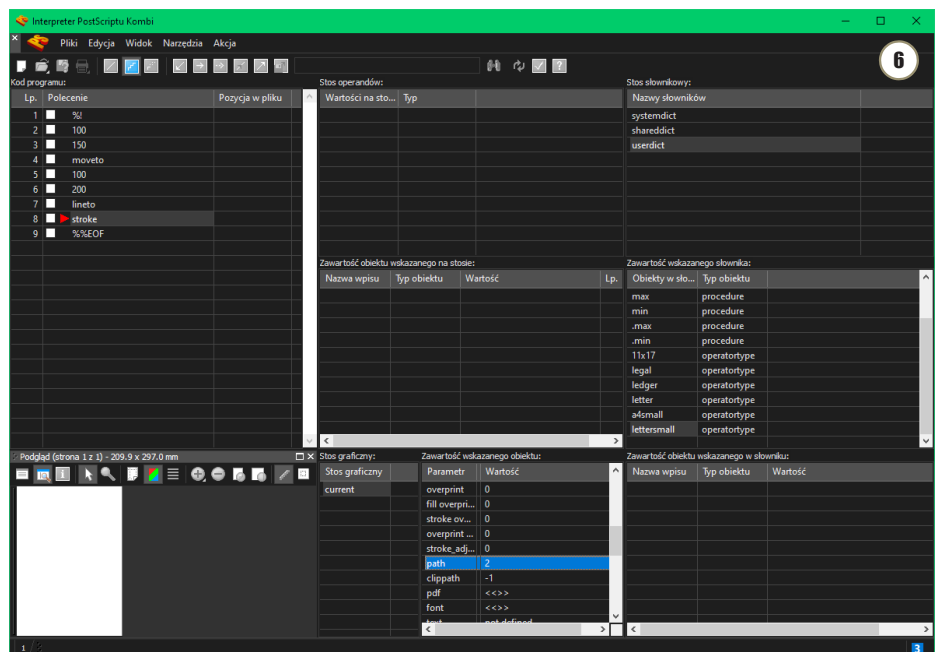
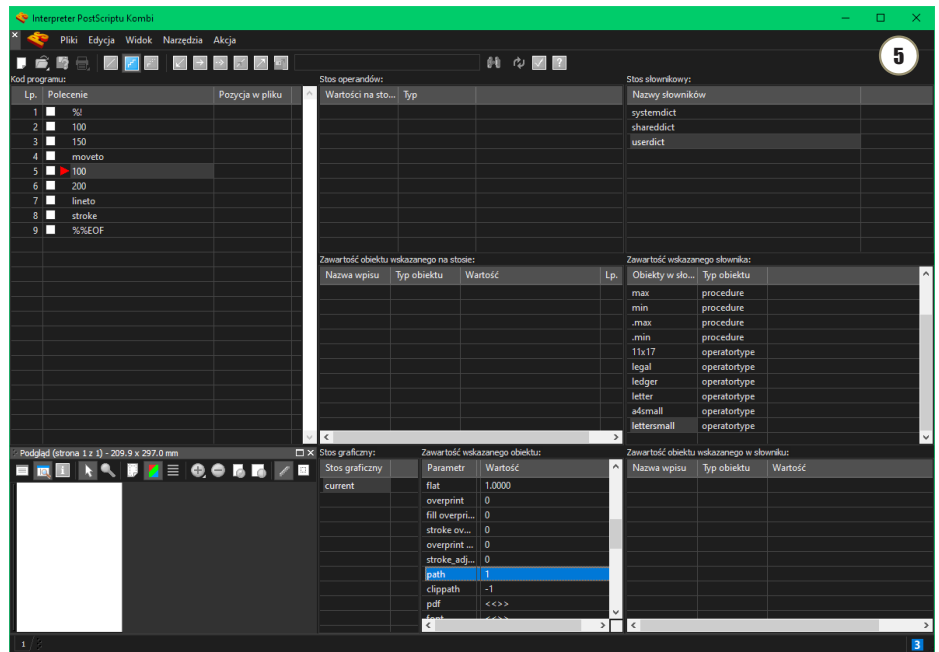
Parametry te zawarte są w **stanie graficznym**. Mamy tu m.in. (omówię te najważniejsze):

- **matrix** – macierz przekształcenia pozwalająca wykonywać operacje geometryczne na stronie (np. obroty, symetrie, itp.);
- **colorSpace** – przestrzeń kolorów. Nas będzie interesowało przede wszystkim to, czy nie jest to **DeviceRGB**. W naszym przykładzie jest **DeviceGray** co oznacza stopnie szarości. Natomiast **DeviceCMYK** oznaczałoby przestrzeń CMYK;
- **color** – ten parametr oznacza nasycenia poszczególnych składowych koloru. W naszym przykładzie – dla **DeviceGray** wartość zero oznacza kolor czarny;
- **linewidth** – oznacza grubość linii;
- **font** – definiuje krój;
- **overprint** – oznacza tryb nadrukowania. Zero oznacza brak nadrukowywania, a jeden oznacza nadrukowywanie.

Podsumujmy. Jeśli wpis na liście **Kod programu** zawiera liczbę lub zmienną, to jest ona przenoszona na stos (lista **Stos operandów**), jeśli natomiast wpis na liście **Kod programu** zawiera operator, to pobiera on wymaganą liczbę parametrów ze stosu (lista **Stos operandów**) i odkłada wynik na ten sam stos. Pozostałe listy widoczne w trybie krokowym pozwalają podglądać zawartość elementów znajdujących się na stosie głównym oraz na dwóch stosach pomocniczych (listy **Stos słownikowy** i **Stos graficzny**).


W trybie krokowym możemy wspomagać się dodatkowymi funkcjami (poza omówionymi wyżej):

- ☑ **Skocz na początek** – program wczyta ponownie analizowany plik i ustawi wskaźnik kolejnej operacji na pozycji 1.
- ☑ **Wykonaj do punktu zatrzymania** – program zostanie wykonany do tej pozycji na liście **Kod programu** przy której ustawimy znacznik (punkty zatrzymania mogą być zapisywane na dysku za pomocą ikony , dzięki czemu po powtórnym otwarciu pliku będą one nadal ustawione).
- ☑ **Wykonaj całość** (wykonuje wszystkie polecenia od bieżącego do końca).
- ☑ **Szukaj** – wyszukuje i podświetla pozycję na liście **Kod programu**, która zawiera tekst znajdujący się w polu edycyjnym za tą ikoną.




Ze względu na fakt, że plik postscriptowy może zawierać dane binarne – nie wszystkie pliki mogą być w trybie krokowym prawidłowo zinterpretowane. Otóż jeśli w pliku znajdują się dane binarne, to za ich interpretację odpowiedzialne są dane znajdujące się wcześniej. Tak więc dopiero po wykonaniu kodu wyznaczającego liczbę bajtów w części binarnej program będzie wiedział ile bajtów powinno być potraktowanych jako dane i w ten sposób zostanie zidentyfikowane miejsce, w którym w pliku znajduje się kolejny token (za częścią binarną). Ponieważ w trybie krokowym tokeny nie są wykonywane na bieżąco, jeśli plik zawiera dane binarne – odnalezienie tokenu za blokiem binarnym nie jest możliwe.



### Tryb zwykły

**Tryb zwykły** (ikona  lub polecenie **Tryb zwykły** w menu **Widok**) pracy interpretera polega na tym, że dane z pliku nie są wstępnie umieszczane na liście **Kod programu**, ale są pobierane i grupowane

w tokeny „w locie” i bezpośrednio po uformowaniu danego tokenu jest on wykonywany. Dzięki temu interpreter może prawidłowo zinterpretować zarówno dane tekstowe jak i binarne. Ten tryb jest „normalnym” trybem pracy i jeśli nie ma potrzeby analizowania pliku (np. w celu znalezienia błędu), to należy interpretować pliki właśnie w tym trybie.


### Tryb krokowy rzeczywisty


**Tryb krokowy rzeczywisty** (ikona ) jest połączeniem dwóch wymienionych wyżej trybów. A więc plik jest analizowany i tokeny są umieszczane na liście **Kod programu**. Ale dane za pozycją bieżącą nie są analizowane, stąd w przypadku danych binarnych będą one zinterpretowane prawidłowo i program będzie mógł prawidłowo przejść do tokenu następującego po danych binarnych.

Ten tryb wykorzystujemy do analizy plików (np. w przypadku poszukiwania błędów) zawierających dane binarne. Wyboru trybu pracy krokowej (ikona  lub ) dokonuje użytkownik (nie ma możliwości

stwierdzenia, czy w pliku są dane binarne przed zinterpretowaniem pliku, tak więc możliwe jest, że przy niesprzyjającym układzie danych w trybie krokowym program wykona niedozwoloną operację – błąd krytyczny). W takiej sytuacji należy uruchomić program ponownie i analizować plik w omawianym tu trybie.

### Okno podglądu

W trakcie analizowania pliku, jeśli powstają dane, które tworzą obraz strony, są one zapisywane w buforze obrazu strony. Zawartość tego bufora może być przeglądana przez użytkownika. W tym celu należy przejść do widoku strony (ikona ) i wysunąć myszką suwak znajdujący się z lewej strony okna.












Zobaczymy wtedy dwie listy (jeśli jesteśmy w trybie krokowym, to okno podglądu może być powiększone za pomocą ikony  na belce tytułowej okienka podglądu – rys. 8). Kolejne pozycje na liście górnej odpowiadają kolejnym wpisom w buforze strony. Z kolei na liście dolnej wyświetlane są dane przypisane pozycji wybranej na liście górnej. Dane te odpowiadają **stanowi graficznemu** omówionemu wcześniej.

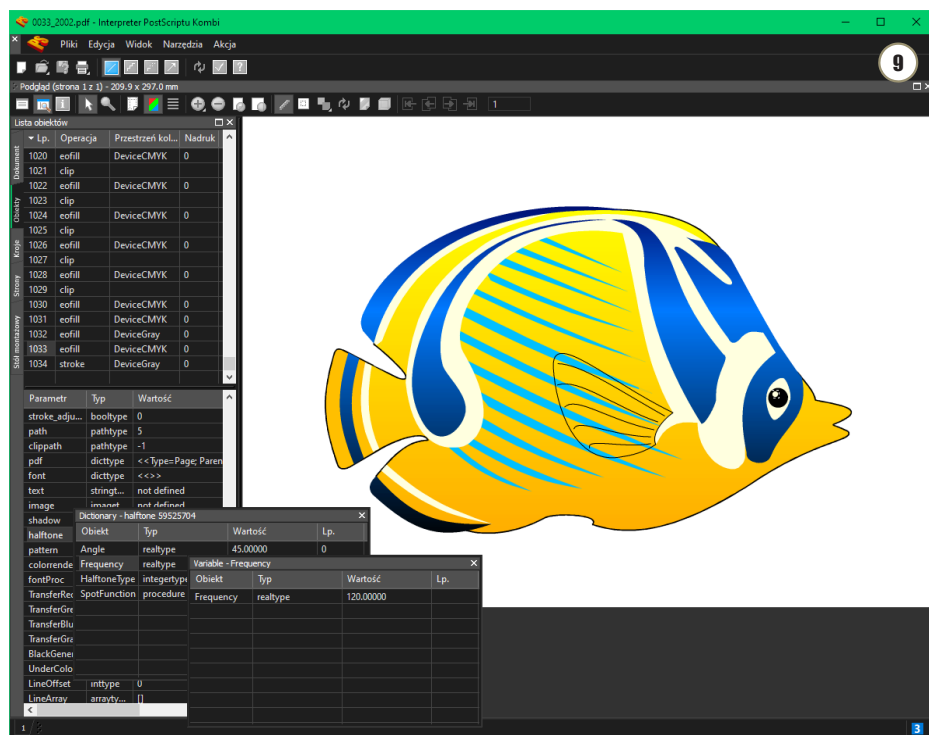
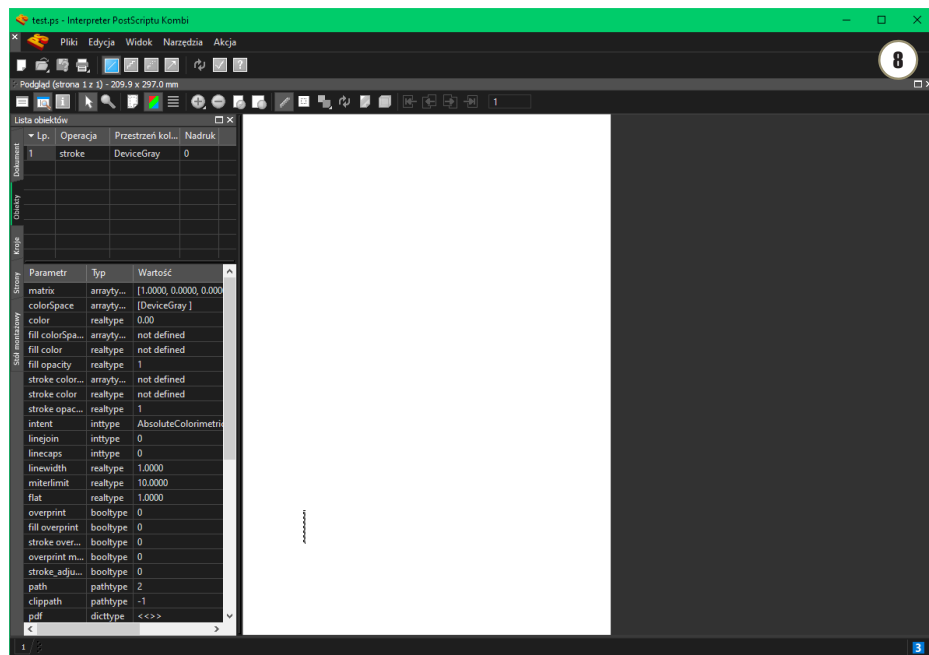
Dzięki możliwości przeglądania danych przypisanych kolejnym obiektom na stronie możemy zorientować się czy dany obiekt ma właściwe atrybuty (np. przestrzeń kolorów, czy atrybut nadrukowania). Ponadto można zobaczyć szczegółowe dane dotyczące wybranego kroju, czy też parametry przypisanej bitmapy.

Jeśli obiekt znajdujący się na liście dolnej jest obiektem złożonym (np. słownikiem, tablicą, bitmapą, krojem, itp.), można podejrzeć jego wewnętrzne składowe. Po kliknięciu obiektu na liście myszką, program otworzy okienko z danymi przypisanymi temu obiektowi. Jeśli i te dane są obiektami złożonymi, to i te obiekty można podejrzeć (rys. 9).

Można kliknąć dwukrotnie w wybraną pozycję na liście górnej, co spowoduje wyświetlenie w oknie tylko tej (wybranej) pozycji bufora. Można też wybrać na liście kilka pozycji (użyj myszki z klawiszem **Ctrl**).

Ponadto w oknie podglądu mamy następujące ikony pomocnicze:





-  Ikona otwiera okno, w którym program wyświetli komentarze zawarte w pliku
-  Ikona włącza tryb wybierania obiektów. W tym trybie można za pomocą myszki wybierać obiekty na stronie, co z kolei powoduje podświetlenie odpowiedniej pozycji na liście górnej.
-  Ikona włącza tryb powiększania. W tym trybie zaznaczamy myszką interesujący nas obszar, który zostanie powiększony tak, by zająć całą powierzchnię okna podglądu.
-  Ikona zwiększa powiększenie o 25%.
-  Ikona zmniejsza powiększenie o 25%.
-  Ikona włącza takie powiększenie, aby cała strona mieściła się w oknie podglądu.
-  Ikona włącza takie powiększenie, aby strona zajęła całą szerokość okna podglądu.
-  Ikona włącza tryb wygładzania krawędzi. Tryb ten spowalnia nieco wyświetlanie danych, ale podnosi jakość wyświetlanego obrazu.
-  Ikona włącza tryb korekcji kolorów na ekranie. Przytrzymanie myszką tej ikony przez czas około pół sekundy otworzy okno definiowania parametrów korekcji.
-  Ikona odświeża obraz w oknie podglądu.
-  Ikona otwiera okno, w którym ustalamy kolor tła (kolor papieru).



Ikona włącza szkieletowy tryb wyświetlania danych.

Pozostałe cztery ikony obsługują

## Pliki wielostronicowe

Program analizuje plik do momentu trafienia na operator **showpage**. W urządzeniach drukujących operator ten powoduje wydrukowanie strony i przejście do analizy następnej strony. Tu operator spowoduje wstrzymanie pracy i dodanie następnej strony (na pasku stanu). Aby przejść do następnej strony należy kliknąć tę stronę (lub użyć ikony ). Aby zmusić program do przeanalizowania całego pliku należy kliknąć ikonę przejścia na ostatnią stronę (ikona ), Ikony  i  powodują odpowiednio przejście na stronę pierwszą lub stronę poprzednią. Ponadto – jeśli plik jest wielostronicowy i pracujemy w trybie ciągłym, za ikonami nawigacji znajdziemy pole edycyjne, w którym możemy edytować numer edytowanej strony.

Dotychczasowy opis wyjaśniał, jak działa PostScript, jak działa interpreter, jak możemy przeglądać atrybuty obiektów, ale nie wyjaśniał

## Jak interpreter współpracuje z Kombi

Interpreter współpracuje z Kombi jako zewnętrzne rozszerzenie, tzn. nie jest on skompilowany razem z Kombi, ale stanowi odrębny program. Takie rozwiązanie zapewnia m.in. niewrażliwość programu głównego (Kombi) na ewentualne błędy krytyczne interpretera. Kombi odwołuje się do interpretera w dwóch opisanych dalej sytuacjach.

1. Kiedy (w Kombi) wykonujemy import do ramki metaplikowej lub wektorowej i wybierzemy filtr mający w nazwie **interpretowany i zewnętrzny**, np. „PostScript interpretowany (filtr zewnętrzny PS)” albo „Grafiki Adobe Illustrator interpretowane (filtr zewnętrzny)” – program przekazuje wybrany plik do interpretera, który otwiera go i **pierwszą stroną** zinterpretowanego pliku zapisuje do pliku tymczasowego jako ramkę Kombi. Następnie Kombi wczytuje ten plik do ramki i operacja importu jest zakończona.
2. Kiedy z menu **Pliki** (w Kombi) wybierzemy funkcję **Otwórz dokument** i z listy filtrów wybierzemy pozycję **Dokumenty postscriptowe** – program przesyła wybrany plik do interpretera PostScriptu, który otwiera go i zapisuje wszystkie strony w pliku tymczasowym jako dokument Kombi. Następnie Kombi otwiera ten tymczasowy dokument.

W obu wypadkach przed przekazaniem pliku do interpretera Kombi otwiera okno z opcjami importu (rys. 10).

Z rozwijanej listy **Poziom PostScriptu** wybieramy poziom (1, 2 lub 3) z jakim interpreter będzie interpretował plik. Generalnie – należy interpretować pliki wybierając poziom 3, ale jeśli chcemy sprawdzić, czy plik zinterpretuje się na urządzeniu z poziomem PostScriptu 1 lub 2 – to możemy zmienić poziom języka na taki, który nas interesuje.

Rozwijana lista **Format arkusza** i sąsiednie pole wyboru orientacji są do-

stępne tylko dla importu całych dokumentów (nie zaś dla importu pliku postscriptowego do ramki) i ustalają format i orientację stron w dokumencie Kombi, który zostanie otwarty w związku z otwarciem pliku postscriptowego. Jeśli znacznik **Nadpisz format danymi z pliku** jest zaznaczony, to format stron w otwieranym dokumencie będzie ustalony na taki, jaki zapisany jest w pliku (oczywiście pod warunkiem, że odpowiednie dane znajdują się w otwieranym pliku).

Pole wyboru **Raport z interpretacji pliku** ustala zachowanie programu po zinterpretowaniu pliku.

**Opcja domyślna – to Wyświetlaj, jeśli są błędy.** W tym stanie – jeśli w czasie interpretowania pliku pojawią się błędy, po zakończeniu pracy interpreter wyświetli odpowiedni komunikat informujący nas o prawdopodobnej przyczynie kłopotów. Dla pozostałych opcji – działanie interpretera jest zgodne z opisem opcji (tzn. albo komunikat nie jest wyświetlany nigdy, nawet w przypadku wystąpienia błędów), albo zawsze – nawet w przypadku bezbłędnej interpretacji.

Ponadto – program może niezależnie od stanu jakiegokolwiek opcji otworzyć okno z prośbą o przesłanie interpretowanego pliku do producenta. Prośba podyktowana jest faktem, że niektórych operatorów nie można było oprogramować z powodu braku dostępu do odpowiedniej liczby plików testowych, w których dany operator byłby wykorzystywany. Stąd – jeśli użytkownik będzie miał taki plik, proszę o przesłanie go na adres [office@3n.com.pl](mailto:office@3n.com.pl).

Warto zwrócić ponadto uwagę na następujące sprawy związane z komunikacją programu Kombi z interpreterem PostScriptu:

1. Określenie **interpretowany** w nazwie filtra nie identyfikuje jednoznacznie omawianego tu interpretera. W samym Kombi był i jest wbudowany uproszczony interpreter, który pozwala importować część wektorową pliku do ramki wektorowej.
2. Podobnie określenie **zewnętrzny** w nazwie filtra może oznaczać filtr importujący pliki postscriptowe jako pliki dowiązane.
3. Filtry używające tego interpretera mają w nazwie oba określenia, czyli **interpretowany zewnętrzny**.
4. Jeśli z jakichś powodów plik \*.eps lub \*.ai zawiera kilka stron i chcemy go otworzyć jako dokument w Kombi, to należy otworzyć ten plik ręcznie w interpreterze, a następnie zapisać go jako dokument Kombi (odpowiednia funkcja znajduje się w interpreterze w menu **Pliki**). Podobnie można otworzyć w interpreterze wielostronicowy dokument postscriptowy i zapisać z poziomu interpretera określoną stroną jako ramkę Kombi.

5. Pliki zinterpretowane nie mogą być z natury rzeczy dowiązane. Jeśli chcemy, aby plik postscriptowy pozostał dowiązany należy użyć filtra z określeniem **nieinterpretowany**.

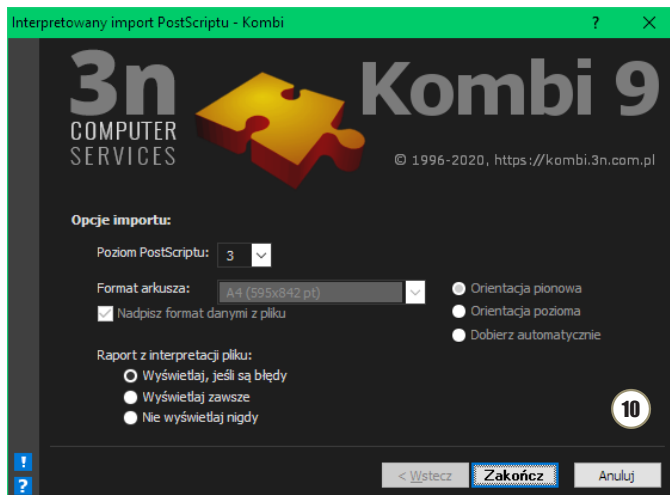
6. Interpretowanie pliku zachodzi w innym procesie, tak więc nie ma możliwości, aby błąd krytyczny interpretera spowodował błąd w programie głównym, jeśli jednak program ulegnie pozornemu zawieszeniu należy użyć klawisza **Esc**, co spowoduje przerwanie pętli, w której program oczekuje na wyniki z interpretera.





Poza omówionymi wyżej najważniejszymi opcjami i funkcjami – mamy w interpreterze szereg opcji i funkcji mniej istotnych, do których mamy dostęp poprzez

## Główne menu okna interpretera

Menu główne programu zawiera następujące grupy, a w grupach – wymienione dalej pozycje:

- **Pliki.** Grupa zawiera następujące pozycje:
  - **Nowy dokument.** Funkcja zamyka otwarty dokument (jeśli jakiś dokument jest otwarty). Dalsze działanie funkcji **Nowy dokument** jest uzależnione od trybu pracy, w którym program się znajduje)
    - **Otwórz** – funkcja otwiera okno wyboru pliku, w którym użytkownik wybiera otwierany plik. Obsługiwane formaty, to: \*.ps (plik postscriptowy), \*.eps (encapsulated postscript), \*.ai (grafiki Adobe Illustrator) oraz \*.pfb (krój Type1). Jeśli użytkownik wybierze plik i zamknie okno wyboru pliku przyciskiem **OK**, program zamyka aktualnie otwarty dokument (jeśli jakiś jest otwarty), a następnie otwiera wskazany dokument. Tryb pracy interpretera nie jest zmieniany. Jeśli aktywny jest tryb ciągły, to interpreter natychmiast po otwarciu przechodzi automatycznie do interpretowania otwartego pliku. W przeciwnym wypadku, program po otwarciu pliku czeka na akcję użytkownika.
    - **Zapisz jako.** Funkcja pozwala zapisać dokument na dysku. Dostępne formaty zapisywanych plików zależą od typu otwartego dokumentu oraz rodzaju operacji, które wykonałimy:
      - PDF – ten format jest dostępny, jeśli otworzyliśmy dokument typu PDF i dokonaliśmy w nim jakichś zmian (np. wykonaliśmy impozycję).
      - KMD – (jest to format własny programu Kombi). Ten format jest dostępny zawsze i zapisywany jest w nim aktualny stan zinterpretowanego dokumentu (jeśli np. zostały zinterpretowane dwie strony ze 100-stronicowego dokumentu, to w pliku KMD znajdują się tylko dwie zinterpretowane strony).
      - KMF – (jest to format własny ramki programu Kombi) – funkcja zapisuje aktualną zawartość bufora strony jako ramkę Kombi. Aby funkcja była dostępna, w buforze strony musi być przeanalizowany przynajmniej jeden obiekt.
      - PS – ten format jest dostępny, jeśli pracujemy w innym trybie niż tryb zwykły (dotyczy to np. programów pisanych przez programistów w edytorze kodu).
  - **Eksportuj wybrany obiekt** – funkcja pozwala wyeksportować fragment otwartego dokumentu. Możliwe do wyeksportowania są następujące obiekty:



- Bitmapy – jeśli na liście obiektów wybrana jest bitmapa, to możemy ją wyeksportować w formacie JPG, PNG, TIF, BMP lub EPS.
  - Ścieżki wektorowe – jeśli na liście obiektów wybrana jest wektorowa ścieżka (wypełnienie, obrys lub ścieżka odcinania), to możemy zapisać ją w formacie EPS.
  - Kroje – jeśli na liście obiektów wybrany jest tekst i ma on przypisany krój w formacie pozwalającym zapisać go jako krój Type 1, to możemy ten krój wyeksportować w formacie PFB.
  - **Wyślij plik pocztą elektroniczną** – funkcja otwiera okno programu pocztowego i dołącza do niego jako załącznik aktualnie otwarty plik.
  - **Drukowanie** – podmenu zawiera dwie pozycje, tj. **Drukuj** i **Konfiguruj**. Wybranie **Drukuj** powoduje wysłanie aktualnie obrabianego pliku do programu drukującego. Może to być np. GhostScript lub program **PrintTo** z pakietu Kombi. Jaki program zostanie użyty do drukowania ustalimy poprzez okno wyboru pliku, które otworzymy poleceniem **Konfiguruj**.
  - **Zarządzanie kolorem** – otwiera okno konfiguracji opcji **Zarządzania kolorem**. Znaczenie i stosowanie opcji zawartych w tym oknie jest omówione w pomocy tego modułu.
  - **Zamknij plik** – zamyka aktualnie otwarty plik i przygotowuje program do interpretowania następnego pliku.
  - **Zamknij program** – zamyka okno programu.
  - **Edycja** – operacje edycyjne zawarte w tym menu dostępne są tylko w trybie krokowym. Są to następujące operacje:
    - **Cofnij** – odwołuje ostatnią operację edycyjną wykonaną na liście **Kod programu**.
    - **Usuń** – usuwa podświetloną pozycję z listy **Kod programu**. Jeśli na liście nie ma podświetlonej pozycji usuwana jest pozycja oznaczona ikoną ▶.
    - **Wstaw** (klawisz **Insert**) – funkcja wstawia pustą pozycję nad podświetloną pozycję na liście **Kod programu**. Jeśli na liście nie ma podświetlonej pozycji, wolna pozycja jest wstawiana nad pozycję oznaczoną ikoną ▶.
    - **Edytuj** (klawisz **F2**) – funkcja otwiera do edycji wybraną pozycję na liście **Kod programu**. Otwarcie to polega na ustawieniu kursora na początku wybranej do edycji pozycji. Po zakończeniu edycji danej pozycji zamykamy edycję klawiszem **Enter**. Spowoduje to automatyczne wstawienie wolnej pozycji za edytowaną pozycją i przeniesienie kursora do tej pozycji. Aby wyjść z trybu edycji bez zatwierdzania wprowadzonych zmian należy użyć klawisza **Esc**.
    - **Zaznacz wszystko** – funkcja wybiera (zaznacza) wszystkie pozycje na liście obiektów w Oknie podglądu.
    - **Odwróć zaznaczenie** – funkcja odwraca zaznaczenie pozycji na liście obiektów w Oknie podglądu.
    - **Otwórz w** – ostatnia pozycja w menu Edycja jest aktywna bez względu na tryb pracy interpretera i pozwala otworzyć obrabiany plik w **Notatniku** i w **GhostScriptcie**. Oczywiście ta druga możliwość jest dostępna tylko po wcześniejszym zainstalowaniu GhostScriptu na naszym komputerze.
  - **Widok** – menu zawiera następujące pozycje:
    - **Tryb zwykły** (ikona ) – włącza podstawowy tryb pracy interpretera. W tym trybie plik jest interpretowany bez możliwości pracy krokowej i podglądania procesu interpretowania.
    - **Tryb krokowy z prekompilacją** (ikona ) – włącza tryb pracy krokowej z możliwością interpretacji danych binarnych.
    - **Tryb krokowy rzeczywisty** (ikona ) – włącza tryb pracy krokowej bez możliwości interpretacji danych binarnych.
- Tryby pracy omówiono we wcześniejszych częściach niniejszego artykułu.
- **Podgląd montażowy**. Polecenie włącza tryb, w którym strony nie są interpretowane. Wyświetlane są tylko numery stron. To znacznie przyspiesza pracę programu i nie wpływa na możliwość wykonywania np. impozycji. Nie jest natomiast możliwe w tym trybie przeglądanie zawartości stron.
  - **Podgląd ekranowy**. Polecenie włącza podgląd ekranowy, w którym strony są interpretowane i wyświetlana jest cała zawartość strony.
  - **Podgląd wybranych obiektów**. Polecenie włącza tryb, w którym można wybrać na stronie te obiekty, które chcemy, aby były wyświetlane (możemy przy tym posiłkować się klawiszami **Shift** i **Ctrl**). Pozostałe (niewybrane obiekty) nie będą wyświetlane.
  - **Format arkusza** – w podmenu wybieramy format oraz orientację arkusza, na którym będzie pracował interpreter. Format ten może być nadpisany danymi znajdującymi się w pliku.
  - **Orientacja widoku** – pozwala obracać widok względem wybranej orientacji arkusza. Np. arkusz papieru może mieć orientację pionową, ale wydruk na nim może być obrócony o 180 stopni co może utrudniać nam analizę zawartości pliku. W takiej sytuacji możemy zmienić orientację widoku przywracając wygodny dla nas widok. W tym samym podmenu możemy włączyć lub wyłączyć **Lustrzane odbicie** widoku oraz **Zresetować widok** (tj. przywrócić wartości domyślne).
- UWAGA!** Ustalona w tym miejscu orientacja (obrót lub odbicie lustrzane) dotyczy tylko widoku, zapisanie dokumentu na tym etapie nie spowoduje tego, że w zapisanym dokumencie zmiany te będą widoczne.
- **Odśwież** – polecenie powoduje powtórne załadowanie pliku z dysku.
  - **Tło przejrzyste** – włącza taki tryb pracy, w którym pole zajmowane przez arkusz papieru nie jest zamalowywane. Pozwala to wyrenderować obraz bez tła (np. do formatu png).
  - **Pokazuj CropBox/BoundingBox** (CropBox to prostokąt wyznaczający format dokumentu po odcięciu marginesów, z ang. *crop* znaczy *przyciąć*. BoundingBox, to najmniejszy prostokąt otaczający wszystkie obiekty na stronie). Włączenie opcji powoduje zaznaczenie na podglądzie przerywaną linią tego prostokąta.
  - **Pokazuj overprint**. Opcja włącza tryb, w którym pokazywany jest efekt nadrukowywania (overprint).
  - **Wyglądanie krawędzi** (ikona ) – opcja włącza tryb wyglądzania krawędzi w oknie podglądu.
  - **Wyłącz przeźroczystość grup** – opcja pozwala wyłączyć pokazywanie przeźroczystości. Umożliwia to sprawdzenie jak będzie wyglądał wydruk na drukarce nieobsługującej przeźroczystości.
  - **Jakość zgrubna** – włączenie tej opcji zwiększa szybkość wyświetlania obiektów na ekranie kosztem pogorszenia jakości odwzorowania szczegółów.

# KOMBInowanie

Poradnik użytkownika programu KOMBI

## Nr 2 (4/2019)

Gazeta wychodzi kiedy chce

Wyłączny sponsor:

**3N**

75-814 Koszalin




ul. Zakole 20/7

kombi.3n.com.pl







e-mail: office@3n.com.pl

Biuletyn złożony za pomocą  
pakietu  
KOMBI v. 9.0

Portable Dokument Format  
wygenerowano rozszerzeniem  
programu KOMBI v. 9.0

- **Podgląd szkieletowy** (ikona ) – opcja włącza pokazywanie grafik wektorowych i tekstu w oknie podglądu w trybie szkieletowym.
- **Kolor strony** (ikona ) – funkcja otwiera okno, poprzez które ustalamy kolor papieru (czyli kolor tła w oknie podglądu).
- **Następna strona, Poprzednia strona, Pierwsza strona i Ostatnia strona** – polecenia pozwalają nawigować po dokumencie. Ich znaczenie opisano w dziale **Dokumenty wielostronicowe**.
- **Lupa** (ikona ) – polecenie włącza lupę, której działanie opisano w dziale **Okno podglądu**.
- **Pasek narzędziowy** – opcja włącza wyświetlanie paska z narzędziami pod menu głównym programu.
- **Pasek stanu** – opcja włącza wyświetlanie paska stanu na dole okna głównego programu.
- **Narzędzia** – podmenu zawiera opcje sterujące pracą interpretera:
  - **Poziom języka** – opcja ustala na jakim poziomie interpreter będzie analizował pliki. Dostępne poziomy, to 1, 2 i 3.
  - **Narzędzia dla programistów** – opcja włącza szereg dodatkowych funkcji wspomagających programowanie w PostScriptcie. Tryby pracy omówione w tym biuletynie będą dostępne po włączeniu tej właśnie opcji. W standardowej konfiguracji opcja ta jest wyłączona i dostępny jest wyłącznie **tryb zwykły** interpretowania plików.
  - **Tryb bezpośredni** – polecenie wykonuje operator **executive**, który wprowadza nas w bezpośredni tryb pracy interpretera. W trybie tym pojawi się dodatkowe okienko z kursorem, w którym możemy wpisywać polecenia języka PostScript, które po zatwierdzeniu klawiszem **Return** będą natychmiast wykonywane. Z trybu **executive** wychodzimy poleceniem **quit**.
  - **Kroje** – podmenu zawiera trzy pozycje. Pozycja **Konfiguruj** otwiera okno wyboru plików, poprzez które wskazujemy katalog, z którego interpreter będzie czerpał kroje. Pozycja **Testuj** wykonuje specjalny testowy program post-

scriptowy odwołujący się do standardowych krojów interpretera. Jeśli plik zinterpretuje się bezbłędnie, oznacza to, że kroje zainstalowane są prawidłowo. Pozycja **Pomoc** otwiera dodatkowe okno pomocy, w której opisano procedurę instalowania krojów. Procedurę tę opiszę w dalszej części niniejszego artykułu.

- **Akcja** – podmenu zawiera funkcje obsługujące tryby krokowe interpretera. Funkcje te w większości wypadków zostały szczegółowo omówione wcześniej, tam też odsyłam po ewentualne szczegóły. W menu znajdziemy następujące pozycje:
  - **Na początek** (ikona ) – powoduje przeskoczenie na początek analizowanego pliku.
  - **Jeden krok** (ikona ) – powoduje wykonanie kolejnego polecenia w trybie krokowym.
  - **Jeden krok bez wchodzenia do procedur** (ikona ) – działa jw., ale jeśli kolejny krok powodowałby wykonanie procedury, to jest ona wykonywana w całości (bez analizy krokowej).
  - **Do punktu zatrzymania** (ikona ) – wykonuje program do najbliższego punktu zatrzymania.
  - **Wykonaj całość** (ikona ) – wykonuje program do końca lub do najbliższego operatora **showpage**.
  - **Wczytaj punkty zatrzymania** – wczytuje zapisane wcześniej punkty zatrzymania.
  - **Zapisz punkty zatrzymania** (ikona ) – zapisuje na dysku punkty zatrzymania. Punkty zatrzymania są zapisywane w pliku o nazwie związanej z nazwą obrabianego dokumentu, stąd każdy obrabiany plik może mieć zapisane niezależnie punkty zatrzymania przypisane do niego.
  - **Wyczyść punkty zatrzymania** – usuwa z listy **Kod programu** wszystkie zaznaczone punkty zatrzymania.

Po omówieniu interfejsu programu, pora na jego skonfigurowanie. Największej uwagi wymaga:

## Konfigurowanie krojów

Zgodnie z dokumentacją – każdy interpreter PostScriptu powinien mieć wbudowane następujące kroje:

- AvantGarde-Book
- AvantGarde-BookOblique
- AvantGarde-Demi
- AvantGarde-DemiOblique
- Bookman-Demi
- Bookman-DemiItalic
- Bookman-Light
- Bookman-LightItalic
- Courier
- Courier-Bold
- Courier-BoldOblique
- Courier-Oblique
- Helvetica
- Helvetica-Bold
- Helvetica-BoldOblique
- Helvetica-Narrow-Bold
- Helvetica-Narrow-BoldOblique
- Helvetica-Oblique
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- NewCenturySchlbk-Roman
- NewCenturySchlbk-Bold
- NewCenturySchlbk-BoldItalic
- NewCenturySchlbk-Italic
- Palatino-Roman
- Palatino-Bold
- Palatino-BoldItalic
- Palatino-Italic
- Symbol
- Times-Roman
- Times-Bold
- Times-BoldItalic
- Times-Italic

- ZapfChancery-MediumItalic
- ZapfDingbats

Wymienionym wyżej krojom mogą odpowiadać dowolne pliki dyskowe (pochodzące od niezależnych producentów), ale należy zadbać o ich prawidłowe związanie z interpreterem. Związanie to odbywa się poprzez dwa omówione niżej pliki.

Plik pierwszy – **fonts.ini** znajduje się w katalogu programu (**..kombi\wtyczki\ps\_int**) i zawiera wymienione wyżej nazwy wymaganych krojów. Ten plik jest dostarczany przez producenta i nie należy go modyfikować.

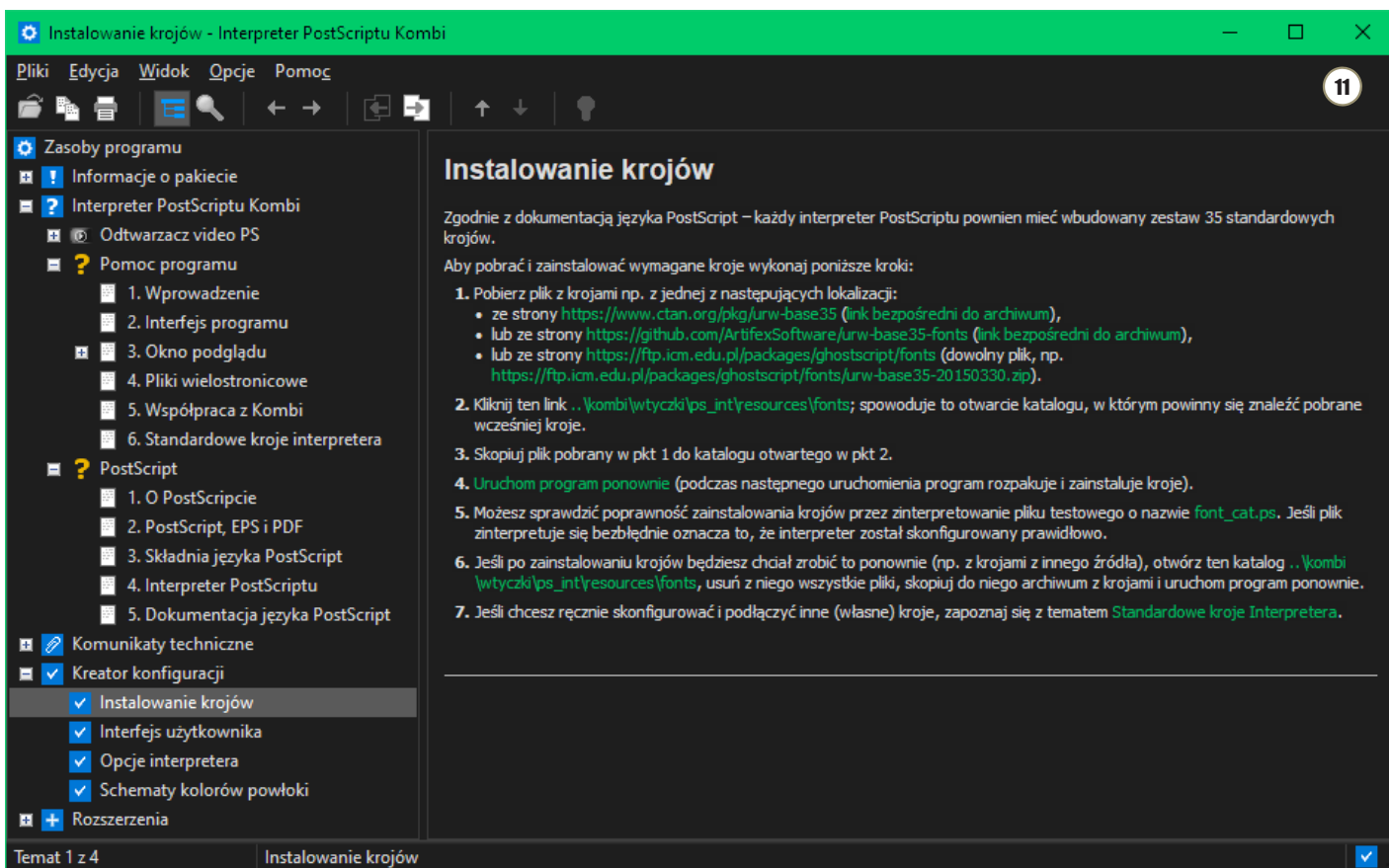
Drugi plik niezbędny do poprawnej pracy programu, to plik **fontmap.ini**, który powinien znajdować się w tym samym katalogu, w którym znajdują się wykorzystywane kroje. Plik fontmap.ini to plik tekstowy, który ma następującą składnię:

```
nazwa-kroju=nazwa-dyskowa
```

gdzie: **nazwa-kroju**, to jedna z 35 wyżej wymienionych nazw, a **nazwa-dyskowa**, to nazwa pliku dyskowego (bez rozszerzenia). Obsługiwane rozszerzenia nazw plików, to **pfb** i **t1**. Program automatycznie doda właściwe rozszerzenie.

Plik **fontmap.ini** można utworzyć samodzielnie i przypisać wymaganym krojom (tym z listy 35 wymaganych krojów) dowolne własne kroje. Jeśli jednak pobierzemy jeden z proponowanych i dostępnych na zasadzie wolnej licencji pakietów, to plik ten zostanie utworzony automatycznie. Kilka przykładowych odnośników do stron zawierających wymagane kroje podano w programie (rys. 11).

Aby sprawdzić poprawność zainstalowania krojów należy zinterpretować plik testowy o nazwie **font\_cat.ps** (menu **Narzędzia** → **Kroje** → **Testuj**). Jeśli plik zinterpretuje się bezbłędnie (rys. 12) oznacza to, że interpreter został skonfigurowany prawidłowo. □





font\_cat.ps - Interpreter PostScriptu Kombi

12

Pliki Edycja Widok Narzędzia Akcja

Podgląd (strona 1 z 1) - 209.9 x 296.7 mm

Lista obiektów

Lp.	Operacja	Przestrzeń kol...	Nadruk
22	text	DeviceGray	0
23	text	DeviceGray	0
24	text	DeviceGray	0
25	text	DeviceGray	0
26	text	DeviceGray	0
27	text	DeviceGray	0
28	text	DeviceGray	0
29	text	DeviceGray	0
30	text	DeviceGray	0
31	text	DeviceGray	0
32	text	DeviceGray	0
33	text	DeviceGray	0
34	text	DeviceGray	0
35	text	DeviceGray	0
36	text	DeviceGray	0

Parametr	Typ	Wartość
matrix	arrayty...	[1.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000]
colorSpace	arrayty...	[DeviceGray ]
color	realtype	0.00
fill colorSpa...	arrayty...	not defined
fill color	realtype	not defined
fill opacity	realtype	1
stroke color...	arrayty...	not defined
stroke color	realtype	not defined
stroke opac...	realtype	1
intent	inttype	AbsoluteColorimetric
linejoin	inttype	0
linecaps	inttype	0
linewidth	realtype	1.0000
miterlimit	realtype	10.0000
flat	realtype	1.0000
overprint	booltype	0
fill overprint	booltype	0
stroke over...	booltype	0
overprint m...	booltype	0
stroke_adju...	booltype	0
path	pathtype	6
clippath	pathtype	0

Standard PostScript fonts

- AvantGarde-Book
- AvantGarde-BookOblique
- AvantGarde-Demi**
- AvantGarde-DemiOblique
- Bookman-Demi**
- Bookman-DemiItalic**
- Bookman-Light
- Bookman-LightItalic
- Courier
- Courier-Bold**
- Courier-BoldOblique**
- Courier-Oblique
- Helvetica
- Helvetica-Bold**
- Helvetica-BoldOblique**
- Helvetica-Narrow-Bold**
- Helvetica-Narrow-BoldOblique**
- nstringval--
- Helvetica-Narrow
- Helvetica-Narrow-Oblique
- NewCenturySchlbk-Roman
- NewCenturySchlbk-Bold**
- NewCenturySchlbk-BoldItalic**
- NewCenturySchlbk-Italic
- Palatino-Roman
- Palatino-Bold**
- Palatino-BoldItalic**
- Palatino-Italic
- Symbol
- Times-Roman
- Times-Bold**
- Times-BoldItalic**
- Times-Italic
- ZapfChancery-MediumItalic

1 3

# Co i jak ?

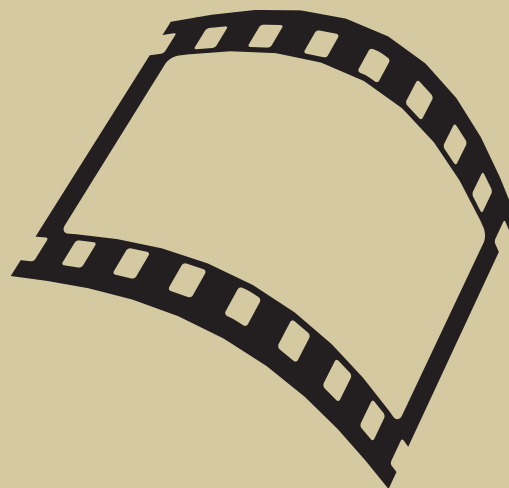
**P**oniżej, w paru krokach pokazuję jak zrobić klatkę filmową.  
**Oczywiście wszystko w Kombi**



1. Z bazy grafik **Zwierzęta lądowe** pobieram interesujące mnie zdjęcie. Dobieram właściwe wymiary i umieszczam w ramce z bitmapą.

2. Przygotowuję grafikę wektorową przedstawiającą klatkę filmu.

3. Wykorzystując **Projektor 3D** rzutuję klatkę na trójwymiarową powierzchnię. Wynik umieszczam w nowej ramce wektorowej. Wykonuję kopię tej ramki, rozwiązuję ją i wchodzę do edycji wektorów skąd przenoszę ścieżki wektorowe do schowka własnego.



4. Otwieram wyjściową ramkę wektorową z klatką i usuwam z niej perforację.

5. Podkładam bitmapę pod ramkę wektorową bez perforacji, związuję obie ramki i konwertuję je na nową ramkę rastrową.

6. Bitmapę z pkt 5 rzutuję w **Projektorze K3D** na tę samą powierzchnię kontrolną, na którą wcześniej rzutowałem ramkę wektorową z perforacją.

7. Wykonuję projekcję bitmapy do nowej ramki rastrowej.

8. Wchodzę do edycji ścieżek w bitmapie utworzonej w pkt 7. Kopiuję ścieżki ze schowka. Zaznaczam wszystkie punkty, przesuвам i skaluję tak utworzoną grupę, aby dopasować ją do bitmapy.

