

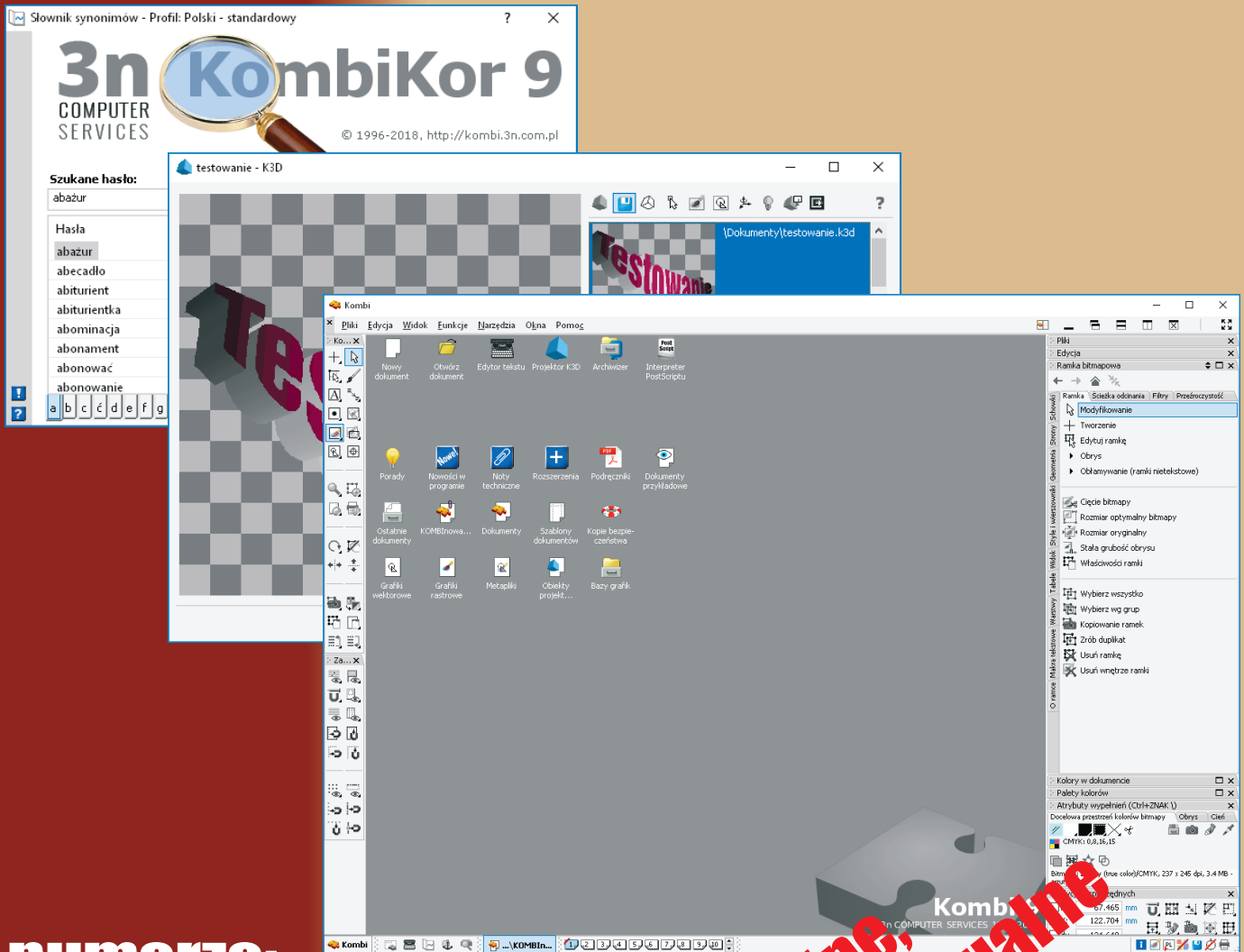


Dla Windows
95/98/ME/NT
2000/XP/7/10

KOMBI nowanie

Poradnik użytkownika programu KOMBI

Numer 1, grudzień 2001 • Gazeta wychodzi kiedy chce



W numerze:

- KOMBI – co to jest ?
- Wymagania sprzętowe i systemowe
- Skróty klawiszowe
- Klawiatura
- Tryby pracy klawiatury
- Kody sterujące
- Style i wierszowniki
- Co i jak ? – karta stałego klienta

UWAGA!
Wydanie Archiwalne,
ale wciąż (prawie) aktualne

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

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

KOMBI – Co to jest ?

W poniższym artykule postaram się odpowiedzieć na fundamentalne pytania związane z programem: czym jest Kombi i czego można się po nim spodziewać ?

Kombi to zintegrowany system komputerowego składu dokumentów. System posiada wbudowane edytory: tekstu, grafiki wektorowej oraz rastrowej i wszystkie te komponenty może połączyć w jeden dokument, pozwalając jednocześnie na ich edycję na każdym etapie składu.

Rdzeniem pakietu jest program **Kombi**, który jest uzupełniany tzw. rozszerzeniami. Rozszerzenia są to zewnętrzne programy usługowe lub biblioteki dynamiczne, z których program może opcjonalnie korzystać. Rozszerzenia nie są niezbędne do pracy programem, ale istotnie poprawiają jego funkcjonalność.

Program obsługuje dokumenty wielostronicowe z dowolną liczbą stron bazowych. Każda strona może mieć przypisaną inną stronę bazową, a także dowolną makietę, czyli układ linii pomocniczych wspomagających tworzenie strony. Ponadto możemy definiować linie pomocnicze globalne, czyli wspólne dla wszystkich stron oraz lokalne – widoczne na danej stronie.

Technika pracy w programie polega na używaniu ramek. Program dostarcza kilka typów ramek. Typ ramki decyduje o tym, jakie dane mogą być w niej przechowywane i jakie operacje mogą być na tych

danych wykonywane. Typ ramki ustalamy przed jej założeniem. Można zmienić typ ramki po jej utworzeniu, ale wiąże się to z utratą zawartych w niej danych. Można natomiast konwertować niektóre typy ramek w inne, ale ta operacja wiąże się zawsze z utworzeniem nowej ramki – nowego typu.

W wersji 8.0 program dysponuje następującymi ramkami: tekstową, wektorową, bitmapową, metaplikową, z kształtem, z paserami, do drukowania i grupą ramek.

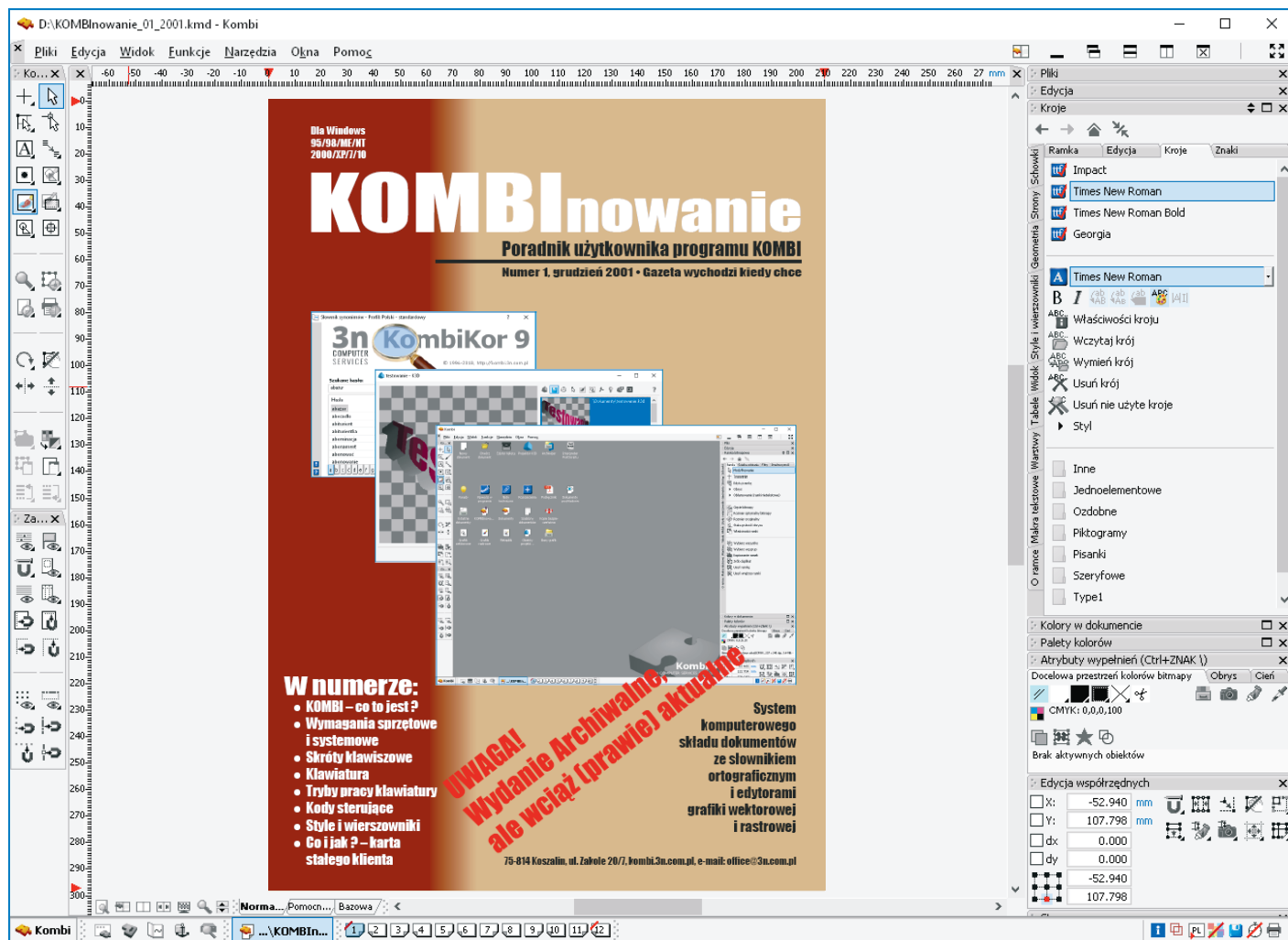
A Ramka tekstowa – służy do przechowywania i składania tekstu. Tekst można importować z dysku, ze schowka lub można wprowadzić ręcznie – z klawiatury. Tekst w pakiecie Kombi jest obsługiwany unikodowo, tzn. każdy znak jest pamiętany na dwóch bajtach, dzięki czemu możliwe było zrezygnowanie ze skryptów, a każdy znak zawarty w kroju jest jednakowo dostępny. Mimo to, w pakiecie istnieją również tzw. strony kodowe, ale tylko celem umożliwienia współpracy z systemami pracującymi w kodzie ASCII (czyli w standardzie jednobajtowym). Z ramką tekstową współpracuje edytor tekstu – **KombiKor**. W pakiecie występuje on w dwóch postaciach. W programie Kombi – jest

wbudowany (wkompiłowany) w program, zaś poza Kombi – występuje jako program samodzielny. Takie rozwiązanie umożliwia stosowanie KombiKora poza pakietem – do współpracy z innymi programami do składu. KombiKor zawiera wbudowany system korekty ortografii i interpunkcji, dzielenia słów na sylaby oraz słownik synonimów. Program zawiera również bogate funkcje adiacji tekstu (np. badanie sparowania cudzysłówów i nawiasów). Mechanizmy wbudowane w słownik pozwalają szukać i wymieniać tekst z uwzględnieniem odmiany wyrazów. Dostępne dla użytkownika są okna dialogowe umożliwiające korygowanie i wprowadzanie nowych słów do słownika.

Tekst w ramach tekstowych może być wyróżniany do wspólnych dla całego dokumentu liniuszków, może oblatywać inne ramki, np. z grafiką, może być również układany na dowolnej ścieżce. W ramce tekstowej można kotwiczyć inne ramki, numery, a także przypisy. Do ramki tekstowej można zaimportować tekst w formatach: ASCII, Unicode, PolType, Rich Text Format (z częściowym zachowaniem atrybutów oraz przypisów), Worda, Write'a. Z ramki tekstowej można wyeksportować tekst w formatach: ASCII, CTX (tekst Calamusa), Rich Text Format, Unicode. Tekst w ramach tekstowych może być wyświetlany za pomocą krojów TrueType, Type 1, OpenType oraz CFN (kroje Calamusa). Kroje nie muszą być zainstalowane w systemie – można je pobierać z dowolnego katalogu, nawet z CD-ROM-u.

Okno programu Kombi v. 9.0.

Konfiguracja: **Kombi 9 standardowy**.



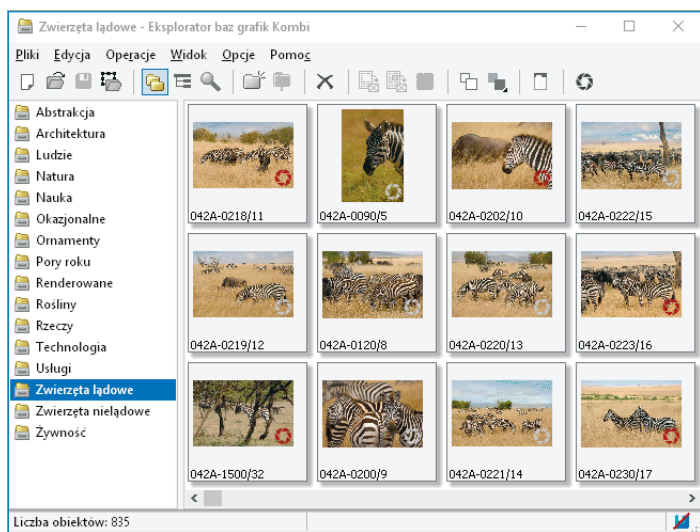
Ramka wektorowa – służy do zarządzania grafikami wektorowymi, które pamiętane są w postaci tzw. **ścieżek** budowanych za pomocą dwóch typów linii, tj.: odcinków i krzywych Béziera. Kombi posiada edytor umożliwiający swobodne manipulowanie punktami węzłowymi oraz kontrolnymi ścieżek. Jako zewnętrzne rozszerzenie dostępny jest generator kodów kreskowych, który generuje kod kreskowy wprost do ramki wektorowej. Do ramki wektorowej można zaimportować pliki w formatach: AI, EPS (tylko część wektorowa), CVG (pliki Calamusa), PLT (pliki w języku HPGL). W tych samych formatach pliki mogą być eksportowane.

Ramka bitmapowa. Ramka może przechowywać bitmapy z: 2, 4, 8, 16, 24 i 32 bitami na piksel. Ramka tego typu poza bitmapą może również mieć przypisany wektorowy kształt, który stanowiąc może tzw. ścieżkę odcinania, która „wycina” z bitmapy żądany kształt. Dla bitmap zawartych w tych ramach można również zadawać kolor przezroczysty, a także przezroczystość (w zadanym stopniu) całej bitmapy. Można edytować zawartość bitmapy za pomocą wbudowanego edytora, w tym również ścieżki odcinania. Bitmapy zawarte w kilku ramach mogą być na siebie nakładane z uwzględnieniem przezroczystości.

Program posiada kilka wbudowanych standardowych filtrów rastrowych (m.in. wyostanie, rozmywanie, itp.), a także obsługuje filtry rastrowe zewnętrzne (rozszerzenia), których liczba sukcesywnie rośnie.

Do ramki rastrowej można zaimportować pliki: BMP (format Windows), GIF (87a i 89a), IMG (GEMImage), JPEG, PCD (Photo-CD), PCX (Paint Brush), PNG, PSD (dokument Photo-Shop – łącznie ze ścieżką odcinania), TGA (Targa), TIFF. Program potrafi również zaimportować wybraną klatkę z pliku video (mpeg, mpg, avi). W zakresie eksportu z ramki rastrowej dostępne są formaty: BMP, GIF, IMG, JPEG, PNG, TGA, TIFF.

Ramki z bitmapami (i nie tylko) mogą być przechowywane w bazach obsługiwanych **Eksplorentem baz grafik**.



Ramka metaplikowa – przechowuje metapliki systemu Windows lub pliki postscriptowe. Nie można edytować ramek tego typu. Można zaimportować metaplik z pliku PS/EPS, AI, WMF lub EMF i wyeksportować go do pliku EMF, PS lub EPS.

Od wersji ósmej pakiet dysponuje własnym Interpreterem PostScriptu.

Ramka z kształtem jest podobna do ramki wektorowej. Różnica polega na tym, że ramka z kształtem ma przypisany typ kształtu i program na tej podstawie jest w stanie po każdej zmianie wymiarów ramki utworzyć w niej nowy kształt. Jeśli np. ustalimy, że ramka z kształtem zawiera prostokąt z zaokrąglonymi narożnikami, o promieniu zaokrąglenia np. 10 mm, to po zmianie wymiarów ramki, program wygeneruje nowy kształt z zachowaniem zadanego zaokrąglenia.

W przypadku ramki wektorowej taka operacja byłaby niemożliwa, a kształt narożników w tak zmienionej ramce nie byłby właściwy. Edycja ramki z kształtem polega jedynie na możliwości wyboru innego kształtu. Nie można importować danych do takiej ramki, można natomiast eksportować dane w takich samych formatach, jak z ramki wektorowej.

Ramka z paserami – przechowuje informacje o rozmieszczeniu paserów, znaczników cięcia oraz innych informacji (np. nazwy wyciągu koloru). Edycja ramki polega na ustaleniu, które elementy i w jakiej konfiguracji mają być w ramce wyświetlane. Nie można importować ani eksportować danych z ramki tego typu.

Ramka do drukowania – jest ramką pomocniczą i umożliwia zaznaczenie fragmentu strony, który będzie wydrukowany. Fragment ten może mieć przypisane inne powiększenie od pozostałej części strony. Powiększenie to może być różne w osi poziomej i pionowej. Nie można importować ani eksportować danych z takiej ramki. Nie istnieje konieczność używania tych ramek, aby uzyskać prawidłowy wydruk. Jest to tylko dodatkowa opcja zwiększająca możliwości programu.

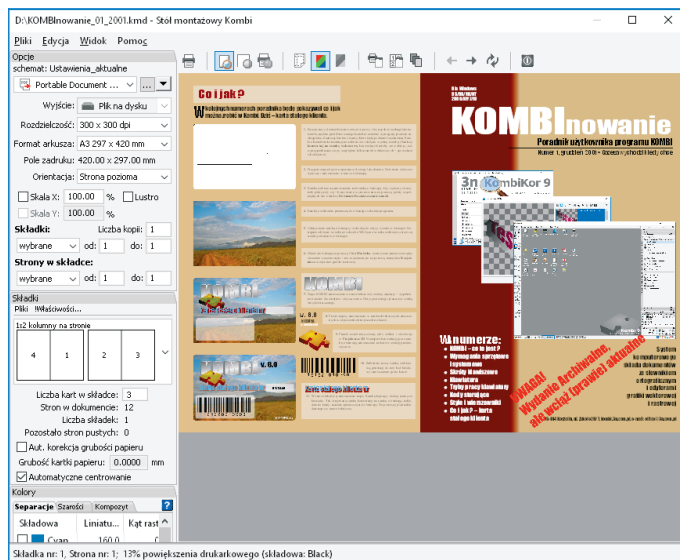
Grupa ramek – każdą liczbę ramek dowolnego typu można łączyć, co spowoduje utworzenie grupy ramek. Nie można edytować zawartości grupy, ale można

wielopozycyjnym schowkiem z podglądem, który umożliwi swobodne kopiowanie i przenoszenie ramek między stronami i dokumentami. Ponadto program wykorzystuje w dużym zakresie technikę „chwyć i upuść”.

Strony dokumentu mogą mieć atrybut „pojedynczej” lub „podwójnej”. Strony podwójne mogą być edytowane w trybie „rozkładówek”, tzn. ramka np. z bitmapą może zająć dwie sąsiednie strony, a w czasie wydruku program zadba o automatyczne rozmieszczenie właściwych fragmentów na właściwych stronach. Tryb ten może być również stosowany na ekranie w czasie edycji.

Program umożliwia drukowanie dokumentów w dwóch trybach. Jeśli urządzenie drukujące posiada interpreter PostScriptu – program generuje plik, który może być do tego urządzenia wysłany. Wykorzystywane są przy tym sterowniki systemowe, tak więc jakość i objętość plików zależy w tym trybie głównie od tych sterowników. Jeśli urządzenie drukujące nie obsługuje PostScriptu lub też chcemy osiągnąć efekty specjalne – możemy włączyć rastrowanie własne. Dla tego trybu, wszystkie obiekty graficzne (łącznie z bitmapami) są rozbarwiane i rastrowane wewnętrznie, a użytkownik ma wpływ na typ, liniaturę i kąt rastra dla poszczególnych składowych. Można również użyć trybu „mieszanego”, a to dzięki możliwości nadawania parametrów rastrowania indywidualnie każdemu obiektowi. Moduł odpowiedzialny za własne rastrowanie obiektów może być również użyty do budowania obiektów na ekranie, co pozwala obejrzeć efekty rastrowania już w czasie składu.

Drukowanie w programie Kombi jest możliwe również z użyciem tzw. stołu montażowego, który pozwala montować złożone strony na dowolny format w dowolnym układzie, w tym z odwracaniem strony o 180 stopni.



zmieniać atrybuty ramek składowych (np. kolory) bez konieczności jej rozwiązania. Nie można eksportować ani importować danych do grupy ramek,

poza jednym wyjątkiem: jeśli grupa zawiera wyłącznie ramki wektorowe – zachowuje się w tym względzie jak zwykła ramka wektorowa.

Wymienione wyżej ramki stanowią podstawowy budulec, z którego powstaje obraz strony. Zarządzanie ramkami jest wspomagane własnym

Kombi umożliwia również wydruk materiału do plików pdf bezpośrednio z programu bez konieczności używania zewnętrznych konwerterów. Pliki pdf generowane przez Kombi są kompresowane i mogą mieć opcjonalnie osadzone kroje zarówno TrueType, OpenType, jak Type 1.

I na koniec trochę statystyki. Na kod źródłowy pakietu składa się ponad 350 plików o łącznej objętości ok. 90 MB. Prace nad pakietem rozpoczęto w połowie 96 roku i trwały bez przerw do roku 2008. Pod naciskiem wiernych użytkowników pakietu, w roku 2017 wznowiono prace i po kilku miesiącach prac zaoferowano nową, dziewiątą wersję pracującą pod systemem Windows 10.

Wymagania sprzętowe i systemowe

Sprawna praca pakietu zależy w dużej mierze od środowiska (zarówno sprzętowego jak i systemowego), w którym program pracuje. Oto krótkie podsumowanie tych zależności.

Pakiet nie ma zbyt wielkich wymagań w zakresie sprzętu. Oczywiście wymagania zależą przede wszystkim od wielkości wykonywanych prac. Sam program w pełnej konfiguracji bezpośrednio po uruchomieniu zajmuje w pamięci około 20 MB. Dla prostego, kilkustronicowego dokumentu bez bitmap, minimalna konfiguracja, w której da się w miarę sprawnie pracować to komputer z zegarem 900 MHz i 64 MB pamięci operacyjnej, choć widziałem w praktycznym zastosowaniu wersję 6.0 programu pracującą na komputerze Pentium II/166 MHz z 32 MB pamięci RAM. Trzeba jednak przyznać, że wydruk strony z takiej konfiguracji trwał nieprzyzwoicie długo.

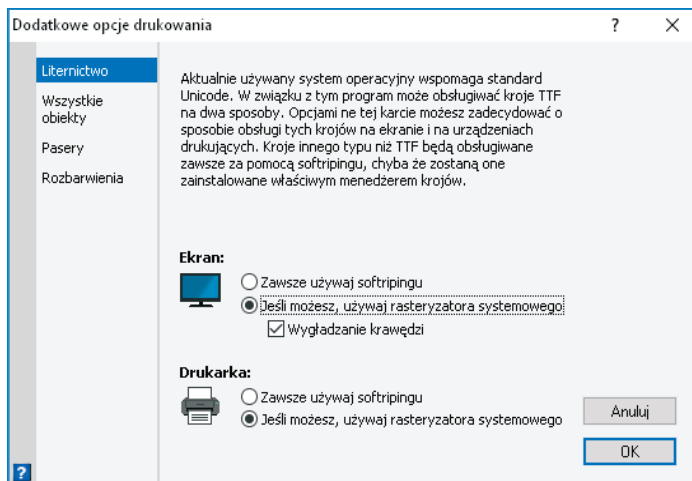
Zapotrzebowanie na pamięć operacyjną rośnie gwałtownie po użyciu w dokumencie bitmap. Trzeba przyjąć, że maszyna winna mieć przynajmniej dwa razy więcej pamięci niż największa z użytych w dokumencie bitmap. Jeśli natomiast chcemy bitmapy obrabiać w programie – to zwróćmy jeszcze uwagę na funkcję „cofnij ostatnią operację”. Dla tej funkcji zadajemy liczbę buforowanych operacji. Przy dużej bitmapie i dużej liczbie pamiętanych operacji rzeczywiście zapotrzebowanie na pamięć może być b. duże. Dla przykładu, ten dokument zajmuje w pamięci około 90 MB.

Moc obliczeniowa jest programowi potrzebna przede wszystkim do obliczenia układu tekstu w ramkach, przy czym takie funkcje jak oblatywanie grafiki tekstem lub unikanie końcowych wierszy akapitu na początku ramki znacznie zwiększają zapotrzebowanie, bowiem program musi dokonać „próbnych dopasowań” i w razie niepowodzenia wykonać obliczenia powtórnie. Sytuację pogarszają długie strumienie tekstu, bowiem zmiany na początku takiego strumienia mogą mieć konsekwencje na jego końcu i program musi to sprawdzić.

Karta graficzna nie ma szczególnego znaczenia na szybkość pracy programu. Chociaż program może pracować w trybie 256 kolorów, to jednak nie jest to zalecane, bowiem w tej sytuacji wszystkie bitmapy są dodatkowo przetwarzane, aby uzyskać symulację większej liczby kolorów. A więc minimum 16 bitów na piksel. Minimalna rozdzielczość, w której da się pracować, to 1024 × 768, ale ze względu na dużą liczbę narzędzi, w tej rozdzielczości komfort pracy nie jest najlepszy. W związku z tym zalecamy 1280 × 1024 lub więcej.

W zakresie wymagań systemowych należy dokonać analizy dwóch rodzin systemów operacyjnych, tzn. linii 95/98/Me i NT/2000/XP. Dla programu Kombi istotne są dwie podstawowe różnice między tymi rodzinami. Po pierwsze rodzina NT obsługuje Unikod i po drugie – obsługuje rozszerzony tryb graficzny. Pierwsza cecha powoduje, że Kombi może rasteryzować systemowo wszystkie

znaki należące do krojów TrueType. W rodzinie 95/98/Me jest to niemożliwe. W związku z tym program pracując pod tymi systemami daje użytkownikowi możliwość wyboru: albo rezygnujemy z unikodu i wtedy program rasteryzuje systemowo, ale tylko jedną, wybraną stronę kodową albo pozostajemy przy unikodzie, ale wtedy program rasteryzuje litery własnym rasteryzatorem. Konsekwencją własnej rasteryzacji są bardzo duże pliki postscriptowe, co wynika z faktu, że do pliku nie jest wysyłany kod litery, ale jej wektorowy obrys. Drugim niekorzystnym zjawiskiem – wynikającym z poprzedniego – jest długi czas transmisji danych. Przejście na rasteryzację systemową jest rozwiązaniem dobrym, ale tylko pod warunkiem używania znaków narodowych należących do wybranej (domyślnie CP1250) strony kodowej. ↓



Drugą właściwość, tj. obsługa (lub jej brak) rozszerzonego trybu graficznego wpływa na pracę stołu montażowego. Przy wybranej opcji obracania stron, tylko rodzina NT robi to systemowo. Rodzina 95 może to zrobić tylko przy użyciu rasteryzatora własnego, z wadami opisanymi wyżej.

Konfiguracja	System operacyjny	Rastrowanie	Objętość pliku	Czas (sek.)
P II MMX 360 MHz, 128 MB RAM	Win 95	Systemowe	1,92 MB	100
P II MMX 360 MHz, 128 MB RAM	Win 2000	Systemowe	1,92 MB	
P II MMX 360 MHz, 128 MB RAM	Win 95	Własne	15,83 MB	480
P II MMX 360 MHz, 128 MB RAM	Win 2000	Własne	15,83 MB	660
Celeron 1,02 GHz, 128 MB RAM	Win 2000	Systemowe	1,92 MB	30
Celeron 1,02 GHz, 128 MB RAM	Win 2000	Systemowe	15,83 MB	320
Celeron 1,02 GHz, 128 MB RAM	Win 98	Systemowe	1,92 MB	65
Celeron 1,02 GHz, 128 MB RAM	Win 2000	Własne	15,83 MB	390
P 4 MMX 1,4 GHz 256 MB RAM	Win 2000	Systemowe	1,92 KB	24
P 4 MMX 1,4 GHz 256 MB RAM	Win 2000	Własne	15,83 MB	165

↓ Zestawienie czasów generowania oraz objętości plików dokumentu pdf tego biuletynu. □

KOMBI nowanie

Poradnik użytkownika programu KOMBI

Nr 1 (12/2001)

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

Podsumowując, wymagany system operacyjny zależy od planowanego wykorzystania programu.

Jeśli planujemy naświetlenia poprzez PostScript i z użyciem stołu montażowego – to niezbędne jest zainstalowanie programu w systemie obsługującym Unikod. Jeśli dodamy tu, że w systemie 2000 również kroje Type 1 (poza TrueType) mogą być obsługiwane bez dodatkowego menedżera krojów, to wybór winien paść właśnie na ten system.

System 95/98/Me może być z powodzeniem stosowany, jeśli podstawowym urządzeniem wyjściowym będzie drukarka niepostscriptowa podłączona wprost do komputera. Wtedy rozmiar plików wyjściowych nie ma istotnego znaczenia, a jeśli dodatkowo w opcjach drukarki ustawimy, aby obiekty graficzne były wysyłane jako bitmapy, to uzyskiwane szybkości drukowania są w zupełności zadowalające.

Wszystko co napisano wyżej odnośnie drukowania ma zastosowanie również do tworzenia plików pdf.

Wszystko co napisano wyżej odnośnie drukowania ma zastosowanie również do tworzenia plików pdf.

Skróty klawiszowe

Jedno z najczęściej zadawanych pytań o zgodność Kombi z innymi programami do składu dotyczy skrótów klawiszowych stosowanych w programie.

Niektóre funkcje programu można wykonywać za pomocą skrótów klawiszowych. Ze względu na sposób ich zdefiniowanie, można je podzielić na dwie grupy:

Grupa pierwsza to te skróty, które są zaprogramowane na stałe – w kodzie programu. Należą do nich operacje wywoływane z menu głównego programu. Znak & oznacza, że sekwencja klawiszy jest dwuczłonowa, np. **Alt P & z** oznacza, że najpierw należy wcisnąć klawisz **Alt** i jednocześnie klawisz **P** co spowoduje otwarcie menu **Pliki**, a następnie wciskamy klawisz **z** co wywoła funkcję **Zapisz**. Znak + oznacza, że dwa (lub więcej klawiszy) należy wcisnąć jednocześnie. Kombinacje klawiszy podane w nawiasach oznaczają skróty alternatywne:

- Alt + Backspace – odwołuje ostatnią operację,
- Alt P & A – **Zapisz dokument jako**,
- Alt P & D – otwiera okno opcji drukowania,
- Alt P & E – eksport z aktywnej ramki,
- Alt P & I – importuje plik do aktywnej ramki,
- Alt P & N – otwiera nowy dokument,
- Alt P & O – wczytuje dokument z dysku,
- Alt P & P – otwiera okno opcji plotowania,
- Alt P & Z – zapisuje dokument na dysku,
- Ctrl + C – wykonuje funkcję **Kopiuuj**,
- Ctrl + V – wykonuje funkcję **Wklej**,
- Ctrl + X – wykonuje funkcję **Wytnij**,
- Ctrl + Z – odwołuje ostatnią operację,
- Ctrl + Insert – wykonuje funkcję **Kopiuuj**,
- Delete – wykonuje funkcję **Usuń**,
- F1 – otwiera pomoc kontekstową programu,
- F12 – otwiera Kreatora konfiguracji,
- Shift + Alt + Backspace – cofa odwołanie ostatniej operacji,
- Shift + Delete – wykonuje funkcję **Wytnij**,
- Shift + Insert – wykonuje funkcję **Wklej**.

Do tej samej grupy należą inne funkcje zaprogramowane wewnątrz programu, a nie dostępne z menu głównego:

- Esc (podczas edycji tekstu) – zmienia tryb pracy klawiatury¹ z trybu wstawiania znaków na tryb komend i odwrotnie,
- Esc (podczas odbudowywania ekranu) – przerywa odbudowywanie ekranu,
- klawisze kursora przesuwają aktywne ramki w odpowiednią stronę o odległość zadaną w Kreatorze konfiguracji (karta klawiatura),
- klawisze kursora – jeśli na stronie nie ma aktywnych obiektów przesuwają widok w oknie z dokumentem w odpowiednim kierunku.
- Ctrl podczas przesuwania ramki myszką powoduje, że ramka może być przesuwana tylko pionowo lub poziomo,
- Ctrl podczas tworzenia lub modyfikowania ramki myszką – włącza lub wyłącza zmiany proporcjonalne ramki,
- Ctrl podczas chwywania ramek myszką do przesuwania lub modyfikowania – włącza tryb kadrowania (tylko, jeśli ramka ma włączone kadrowanie),

- Ctrl podczas tworzenia ramki tekstowej powoduje połączenie jej z ostatnio obrabianą ramką (jeśli ostatnio obrabiana była ramka tekstowa),
- Ctrl przytrzymany w czasie chwywania wnętrza ramki, gdy ta jest kadrowana powoduje przesunięcie wnętrza ramki względem jej obrysu,
- Ctrl przytrzymany w czasie chwywania obrysu ramki, gdy ta jest kadrowana powoduje zmianę wymiarów ramki i jednocześnie zmianą wymiarów jej zawartości (domyślnie dla ramek kadrowanych taka akcja zmienia tylko wymiary obrysu nie zmieniając zawartości),
- Ctrl + F12 włącza/wyłącza obsługę skrótów klawiszowych (funkcja pozwala wyłączyć na chwilę obsługę skrótów, aby umożliwić wprowadzenie niektórych znaków z klawiatury w trybie edycji tekstu),
- Shift podczas chwywania aktywnej ramki myszką powoduje przesunięcie tylko chwyconej ramki (a nie wszystkich aktywnych),
- Shift podczas modyfikowania lub tworzenia ramki myszką – włącza lub wyłącza przyciąganie do zatrząsków,
- Shift łącznie z prawym przyciskiem myszki – uaktywnia przesunięcie kartki w oknie roboczym,
- Shift przytrzymany w czasie zaznaczania ramek myszką („zaciągania”) zmienia tryb wybierania ramek z „wybierz otoczone” na „wybierz przecięte” i odwrotnie,
- Shift łącznie z lewym przyciskiem myszki, gdy ta jest w trybie zadawania powiększenia – pomniejsza obraz o 100%,
- Ta sama operacja bez klawisza Shift – powiększa obraz o 100%,
- Shift podczas obracania ramki za pomocą myszki włącza zatrząski na kątach będących wielokrotnościami kąta równego 15 stopni.

Shift + Esc – włącza lub wyłącza tryb programowania klawiatury.

W ramce tekstowej działają następujące kombinacje klawiszy (poza standardową obsługą klawiatury):

- klawisze kursora łącznie z klawiszami Alt + Ctrl – wprowadzają kod ręcznego kerningu – jedno wciśnięcie klawisza wprowadza przesunięcie litery o 1% wysokości litery,
- klawisze kursora łącznie z klawiszami Alt + Ctrl i Shift – jw. – ale wprowadzone przesunięcie wynosi 10% wysokości litery H,
- klawisze kursora z jednocześnie przytrzymanym klawiszem Shift powodują zaznaczanie fragmentu tekstu,
- klawisze kursora łącznie z klawiszem Ctrl (ale nie w trybie ręcznego kerningu) przesuwają kursor do poprzedniego lub następnego akapitu,
- klawisze PageUp i PageDown łącznie z klawiszem Ctrl ustawiają kursor odpowiednio na początek poprzedniej lub następnej ramki w strumieniu tekstu.
- klawisze Home i End łącznie z klawiszem Ctrl ustawiają kursor odpowiednio na początku i końcu ramki.

- Shift w trybie komend włącza tryb przesuwania akapitu,
- dwuklik myszką z wciśniętym klawiszem Ctrl zaznacza wskazany myszką akapit.

Grupa druga, to domyślnie zaprogramowany układ klawiszy w oknie Eksploratora komend.

W całym programie:

- 1 – blokuje wartość X (w Palecie edycji współrzędnych),
- 2 – blokuje wartość Y (jw.),
- A – Równanie i rozmieszczanie ramek,
- Delete – usuwa wybraną ramkę (ramki) (jeśli w ramce tekstowej jest aktywny kursor - to klawisz Delete usuwa znak. Aby w tej sytuacji usunąć ramkę - możemy użyć kombinacji Ctrl + Delete),
- F – otwiera okno definiowania parametrów strony,
- F2 – otwiera okno przeglądarki dokumentów Kombi,
- F3 – włącza sprawdzanie tekstu w ramkach,
- F4 – włącza edycję ramki (przy braku aktywnych ramek otwiera edytor tekstu),
- F5 – odświeża obraz strony,
- F6 – włącza lub wyłącza zatrząski w Palecie edycji współrzędnych,
- F7 – włącza lub wyłącza tryb pracy – obracanie/pochylenie ramek,
- F8 – wywołuje funkcję edycji kodów sterujących,
- F9 – otwiera arkusz właściwości ramki,
- F11 – włącza tryb pełnoekranowy,
- G – łączy ramki w grupę,
- H – włącza edycję wysokości ramki,
- K – włącza lub wyłącza tryb kadrowania,
- PageDown – przechodzi na następną stronę,
- PageUp – przechodzi na poprzednią stronę,
- TAB (lub Shift + PgUp) – uaktywnia ramkę znajdującą się nad ramką aktywną,
- U – rozwiązuje grupę ramek,
- V – włącza tryb modyfikowania obiektów,
- W – włącza edycję szerokości ramki,
- X – włącza edycję współrzędnej X ramki,
- Y – włącza edycję współrzędnej Y ramki,
- Ctrl + 0 – pokazuje całą stronę,
- Ctrl + A – wybiera wszystkie obiekty,
- Ctrl + B – włącza lub wyłącza wyfłuszczenie,
- Ctrl + D – wykonuje duplikat ramki (z przesunięciem kopii),
- Ctrl + DELETE – usuwa wybraną ramkę (ramki),
- Ctrl + E – eksportuje zawartość ramki,
- Ctrl + F8 – kopiuje kod sterujący (styl lub wierszownik) do schowka,
- Ctrl + I – importuje zawartość ramki,
- Ctrl + K – włącza lub wyłącza pochylenie,
- Ctrl + L – kotwiczy obiekt w tekście,
- Ctrl + M – dołącza dokument z dysku,
- Ctrl + N – otwiera nowy dokument,
- Ctrl + NUM0 – pokazuje całą stronę,
- Ctrl + NUM1 – włącza klawiaturę nr 1,
- Ctrl + NUM2 – włącza klawiaturę nr 2,
- Ctrl + NUM3 – włącza klawiaturę nr 3,
- Ctrl + NUM4 – włącza klawiaturę nr 4,
- Ctrl + NUM5 – włącza klawiaturę nr 5,
- Ctrl + NUM6 – włącza klawiaturę nr 6,
- Ctrl + NUM7 – włącza klawiaturę nr 7,
- Ctrl + NUM8 – włącza klawiaturę nr 8,
- Ctrl + NUM9 – włącza powiększenie na szerokość okna,
- Ctrl + NUM+ (numeryczny plus) – powiększa obraz o współczynnik zadany w oknie ręcznego zadawania powiększenia,

¹ Patrz artykuł **Tryby pracy klawiatury**.

- Ctrl + NUM- (numeryczny minus) – pomniejsza obraz o współczynnik zadany w oknie ręcznego zadawania powiększenia,
- Ctrl + NUM kropka – pokazuje poprzednie powiększenie,
- Ctrl + O – otwiera dokument z dysku,
- Ctrl + P – otwiera okno opcji drukowania,
- Ctrl + PgUP – przechodzi do poprzedniej rozkładówki,
- Ctrl + PgDOWN – przechodzi do następnej rozkładówki,
- Ctrl + Q – wybierz akapit,
- Ctrl + R – otwiera okno kopiowania ramek,
- Ctrl + S – zapisuje dokument na dysku,
- Ctrl + SPACJA – włącza lupę podręczną,
- Ctrl + ŚREDNIK – włącza lub wyłącza pokazywanie elementów pomocniczych na stronie,
- Ctrl + T – otwiera okno opcji plotowania,
- Ctrl + W – otwiera Menedżera krojów,
- Ctrl + [– przesuwa ramkę jeden plan w dół,
- Ctrl +] – przesuwa ramkę jeden plan w górę,
- Ctrl + ZNAK \ – otwiera paletę atrybutów wypełnień,
- Shift + A – rozmieszczanie ramek,
- Shift + F2 – włącza adyustację tekstu w ramkach,
- Shift + TAB (lub Shift + PgDown) – uaktywnia ramkę znajdującą się pod ramką aktywną,
- Shift + F3 – włącza dzielenie tekstu w ramkach,
- Shift + F5 – włącza lub wyłącza tryb pokazywania dwóch stron,
- Shift + F8 – wkleja kod sterujący (styl lub wierszownik) ze schowka w miejsce wskazywane kursorem,
- Shift + F11 – pokazuje lub ukrywa pulpit,
- Shift + F12 – pokazuje lub ukrywa palety,
- Shift + PgUP – wybiera poprzednią ramkę tekstową,
- Shift + PgDOWN – wybiera następną ramkę tekstową,
- Ctrl + Shift + D – wykonuje duplikat ramki (bez przesunięcia kopii),
- Ctrl + Shift + O – otwiera ostatnio obrabiany dokument,
- Ctrl + Shift + Q – wybierz wyraz,
- Ctrl + Shift + U – ustawia standardowe jednostki,
- Ctrl + Shift + V – wkleja specjalnie,
- Ctrl + Shift + [– przesuwa ramkę na ostatni plan,
- Ctrl + Shift +] – przesuwa ramkę na pierwszy plan,
- Ctrl + Shift + PageDown – przeskoczyć na koniec dokumentu,
- Ctrl + Shift + PageUp – przeskoczyć na początek dokumentu,
- Alt + kursor w lewo – przejdź do tyłu,
- Alt + kursor w prawo – przejdź do przodu,
- Alt + I – włącza lub wyłącza pokazywanie kodów sterujących,
- Alt + K – włącza przycinanie zawartości ramki do jej obudowy,
- Alt + U – przewija jednostki w górę,
- Alt + Ctrl + I – włącza lub wyłącza pokazywanie znaków specjalnych,
- Alt + Shift + U – przewija jednostki w dół.

W module wektorowym:

- C – wybiera wprowadzanie odcinków,
- C + R – otwiera okno kopiowania ścieżek,
- O – włącza operacje geometryczne na grupie punktów,
- S – włącza lub wyłącza utrzymywanie styczności w punkcie,

- Z – wybiera wprowadzanie krzywych beziera.

W edytorze tekstu:


- Alt + Ctrl + NUM / – wstawia kod **Wyłącz zaawansowane justowanie**,
- Alt + Ctrl + NUM gwiazdka – wstawia kod **Nadrób wiersz warunkowo**,
- Alt + Ctrl + NUM plus – wstawia kod **Nadrób wiersz**,
- Alt + Ctrl + NUM minus – wstawia kod **Zgub wiersz**,
- Alt + Ctrl + TAB – wstawia kod tabulatora kończącego,
- Alt + Ctrl + ZNAK = – wstawia kod **Połącz akapit z następnym**,
- Ctrl + B – włącza lub wyłącza styl wytłuszczenia znaków,
- Ctrl + E – eksport,
- Ctrl + F – otwiera panel szukania i wymiany tekstu,
- Ctrl + I – import,
- Ctrl + J – wstawia styl kasujący wytłuszczenie i pochylenie znaków,
- Ctrl + K – włącza lub wyłącza styl pochylenia znaków (kursywę),
- Ctrl + Q – zaznacza akapit,
- Ctrl + Shift + Q – zaznacza wyraz,
- Esc – zakończ sprawdzanie,
- F3 – sprawdzanie tekstu,
- F4 – zignoruj błąd,
- Num * – zamyka edytor,
- Num / – przelewa tekst z edytora do ramki i zamyka okno edytora.

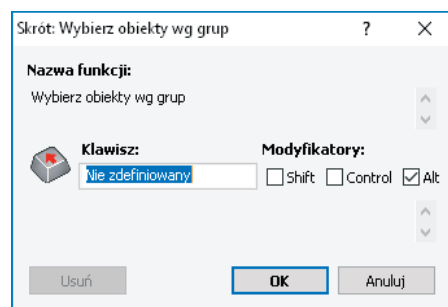
Wymienione wyżej powiązania mogą być przez użytkownika zmieniane.

Aby przypisać (lub zmienić) skrót klawiszowy do funkcji przypisanej ikoncie, należy kliknąć ją prawym przyciskiem myszki i z otwartego menu wybrać pozycję **Przypisz skrót klawiszowy** (jeśli pozycja jest niedostępna, co jest rzadkością w programie, oznacza to, że dana funkcja nie może mieć przypisanego skrótu klawiszowego).

W otwartym w ten sposób oknie ustalamy klawisz skrótu oraz ewentualnie modyfikator.


Aby przypisać (lub zmienić) skrót klawiszowy do pola edycyjnego, należy wybrać tryb programowania klawiatury (Shift + Esc), wskazać żądane pole edycyjne i przycisnąć proponowany klawisz skrótu z ewentualnym modyfikatorem.

Aby przypisać (zmienić lub usunąć) skrót klawiszowy do dowolnej funkcji programu, należy otworzyć okno **Eksploratora komend** (np. z menu **Kombi/Ustawienia**) i wybrać funkcję **Przypisz skrót klawiszowy** z menu podręcznego interesującej nas funkcji. Spowoduje to otwarcie okna definiowania skrótu, w którym zdefiniujemy skrót. 



Skróty klawiszowe są pamiętane w pliku konfiguracyjnym programu. Zmiana pliku konfiguracyjnego (np. po aktualizacji) spowoduje utratę zapamiętanych w nim skrótów.

Aby temu zapobiec, można zapisać zdefiniowane przez siebie skróty do pliku. Zrobimy to poprzez okno **Eksploratora komend** i dalej przez podręczne menu listy – funkcja **Zapisz skróty**. Najbezpieczniej jest nadać zapisywanemu plikowi własną nazwę (inną od standardowej), aby proces aktualizacji programu nie nadpisał tego pliku.

Następnie, po aktualizacji, możemy wczytać plik podobnie jak go zapisywaliśmy. Można również zapisać skróty do pliku tekstowego, co umożliwi wydrukowanie podręcznej „ściągi”. 

Klawiatura




Kontynuując wątek obsługi klawiatury warto zapoznać się z dodatkowymi możliwościami oferowanymi w tym zakresie przez program Kombi.

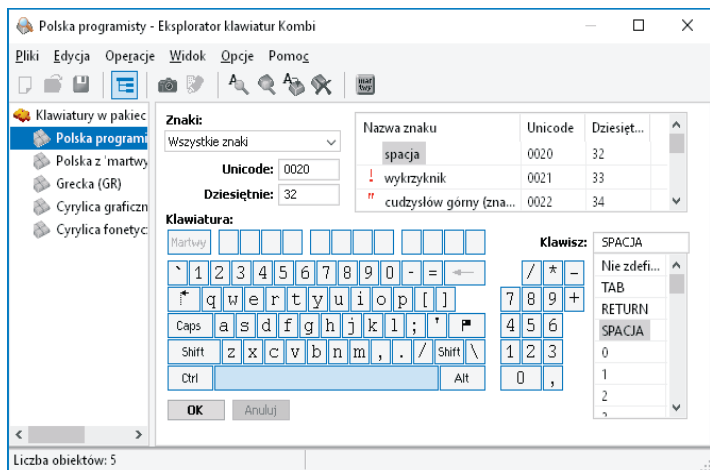
Program Kombi może obsługiwać klawiaturę na dwa sposoby. Sposób pierwszy wykorzystuje systemowe sterowniki klawiatury i jest zgodny ze wszystkimi innymi programami wykorzystującymi te sterowniki. Niestety, poza tą zgodnością, sposób ten ma wadę polegającą na niemożności łatwego wprowadzania znaków specjalnych, takich jak cudzysłowy, pauzy, półpauzy, itp.

Wad tych pozbawiony jest sposób drugi polegający na wykorzystaniu własnych sterowników obsługi klawiatury. Wyboru sposobu obsługi klawiatur dokonujemy w **Kreatorze konfiguracji** na zakładce **Klawiatura**.

Po wybraniu własnej obsługi klawiatury, a jest to stan domyślny dla programu, przechwytytuje on wszystkie odwołania do klawiatury i odszukuje właściwe kody wprowadzanych znaków w zdefiniowanych przez użytkownika tablicach. Tablice

definiujemy w **Eksploratorze klawiatur** (menu **Narzędzia**).

W oknie **Eksploratora klawiatur** możemy powiązać klawisze z kodami Unicode tak, aby możliwe było uzyskiwanie znaków narodowych wprost z klawiatury. Lewy panel okna (otwierany ikoną ) zawiera nazwy załadowanych aktualnie do pamięci definicji klawiatur. Z listy wybieramy tę klawiaturę, którą chcemy zmieniać (jeśli chcemy stworzyć klawiaturę od podstaw, to najpierw musimy utworzyć nową definicję klawiatury za pomocą ikony ). Do nazwy klawiatury przypisany jest skrót, który możemy edytować poprzez menu kontekstowe – pozycja **Zmień nazwę skrótu**. W tym samym menu znajdziemy również funkcję usuwania definicji klawiatury z pamięci oraz funkcję przypisującą skrót klawiszowy danej definicji. 



Prawy panel okna zawiera w górnej części rozwiązaną listę zatytułowaną **Znaki**. Lista ta zawiera wykaz dostępnych zestawów znaków Unicode. Jeśli chcemy np. zaprogramować tylko polskie znaki diakrytyczne, to nie musimy przeglądać całego zestawu dostępnych znaków, możemy wybrać zestaw **Litery polskie** i wtedy tylko te będą wyświetlane na sąsiedniej liście. Z tej listy wybieramy znak, który chcemy przypisać jakiemuś klawiszowi. Kod tego znaku będzie wyświetlany jako kod Unicode oraz dziesiętnie w odpowiednich polach edycyjnych. Użytkownik może wybrać znak również poprzez wpisanie jego kodu w odpowiednie pole.

Następny etap programowania polega na wybraniu klawisza, któremu chcemy przyporządkować wskazany wcześniej znak. Wyboru klawisza dokonamy w polu edycyjnym **Klawisz**. Kiedy je uaktywnimy, pojawi się w nim kursor, ale kiedy będziemy wprowadzali w to pole tekst – program będzie przetwarzał kody wciskanych klawiszy na ich nazwy i one będą wyświetlane w tym polu. Aby jednoznacznie sprecyzować klawisz, wystarczy więc go wcisnąć. Możemy również wskazać klawisz myszką na symbolicznej klawiaturze zobrazowanej w tym oknie poprzez przyciski. Dodatkowo możemy określić tzw. modyfikatory (czyli klawisze modyfikujące działanie klawisza podstawowego). Dozwolone są tu wszelkie kombinacje dostępnych modyfikatorów (a więc Shift, Ctrl, Alt i Caps). Definiując klawisze z modyfikatorem Ctrl należy zwrócić uwagę na następujący fakt. Jeśli do tej samej kombinacji klawiszy przypiszemy znak na klawiaturze, skrót klawiszowy oraz skrót do stylu (wierszownika), to pierwszeństwo ma skrót do stylu (wierszownika), dalej – skrót do funkcji i na końcu – znak przypisany klawiszowi. Skróty klawiszowe do funkcji przypisane bez modyfikatora **Ctrl** w trybie wprowadzania znaków (tzn. przy aktywnym kursorze w ramce tekstowej) nie są uwzględniane (poza skrótami przypisanymi klawiszom funkcyjnym). Dodatkowym modyfikatorem jest specjalny klawisz opisany jako **Martwy**. Klawisz ten stanie się aktywny, kiedy przyciskiem ustalimy, który ze zwykłych klawiszy będzie pełnił rolę „martwego”. Klawisze zdefiniowane dla właściwego modyfikatora **Martwy** będą wysyłały właściwy znak do programu po użyciu sekwencji klawiszy – klawisz „martwy” + wybrany znak. Po określeniu klawisza możemy użyć przycisku . Spowoduje to powiązanie wybranego znaku z wybranym klawiszem, co kończy cykl programowania jednego znaku.

Przyciski: oraz pełnią rolę pomocniczą. Pierwszy z nich wyszukuje klawisz, który jest powiązany z wybranym aktualnie znakiem. Ponieważ

takich klawiszy może być kilka, kolejne wciśnięcia tego przycisku przesuwają nas na kolejny klawisz. Drugi przycisk usuwa powiązanie wybranego klawisza z przypisanym mu znakiem.

Przycisk wyszukuje w zestawie znaków ten znak, który jest przypisany aktualnie wybranemu klawiszowi. Zwróćmy jednak uwagę, że poszukiwania znaku ograniczają się jedynie do wybranego zestawu znaków. Jeśli wybierzemy np. zestaw **litery greckie**, to nie znajdziemy przyporządkowania dla klawiszy, które obsługują litery polskie.

Ikona otwiera z dysku zapisaną tam wcześniej definicję klawiatury. Jest ona dołączana na koniec listy zawierającej nazwy klawiatur.

Ikona zapisuje wybraną (tzn. aktualnie obrabianą) klawiaturę na dysku. Zdefiniowane klawiatury są domyślnie przechowywane w pliku konfiguracyjnym programu. Zaleca się jednak dodatkowe zapisanie zmodyfikowanej klawiatury we własnym formacie (*.key) pod nazwą inną od nazw nadanych przez producenta. Umożliwi to w przyszłości odtworzenie wprowadzonych zmian po ewentualnej aktualizacji programu, która może spowodować nadpisanie pliku konfiguracyjnego oraz plików z definicjami klawiatur.

Ikona zapisuje wybraną (tzn. aktualnie obrabianą) klawiaturę na dysku. Zdefiniowane klawiatury są domyślnie przechowywane w pliku konfiguracyjnym programu. Zaleca się jednak dodatkowe zapisanie zmodyfikowanej klawiatury we własnym formacie (*.key) pod nazwą inną od nazw nadanych przez producenta. Umożliwi to w przyszłości odtworzenie wprowadzonych zmian po ewentualnej aktualizacji programu, która może spowodować nadpisanie pliku konfiguracyjnego oraz plików z definicjami klawiatur.

Tryby pracy klawiatury

W różnych stanach programu oczekujemy różnych reakcji na wprowadzane przez nas dane z klawiatury. W związku z tym wprowadzono tzw. tryby pracy klawiatury, które omawiam poniżej.

W wersji 8.0 programu dostępne są następujące tryby pracy klawiatury:

- Tryb wstawiania znaków – powoduje wprowadzanie znaków w miejscu wskazywanym przez kursor w ramce tekstowej bez usuwania innych znaków.
- Tryb wymiany znaków – działa jak tryb poprzedni, ale wprowadzenie nowego znaku powoduje usunięcie znaku znajdującego się bezpośrednio przed kursorem.
- Tryb komend – powoduje przesyłanie danych z klawiatury nie do aktywnej ramki tekstowej, a do głównego okna programu, co pozwala na wywoływanie skrótów do funkcji za pomocą klawiatury.
- Tryb programowania – , który służy do programowania skrótów do funkcji.
- Tryb ręcznego kerningu – . Ten tryb ma zastosowanie tylko w czasie edycji ramek tekstowych. W tym trybie kombinacje klawiszy Ctrl + klawisze kursora oraz Ctrl + Shift + klawisze kursora wprowadzają w tekst tzw. ręczny kerning, czyli pozwalają na ręczną korektę światła między literami.

Ikona kopiuje wszystkie klawisze z wybranymi aktualnie modyfikatorami do schowka.

Ikona wkleja zawartość schowka na wszystkie klawisze z wybranymi aktualnie modyfikatorami. Wspomniana już wcześniej ikona otwiera nową (czystą) definicję klawiatury.

Producent dostarcza pięć zdefiniowanych klawiatur. Są to:

PL – klawiatura polska z polskimi znakami otrzymywanymi za pomocą klawisza **Alt**. Dodatkowo z klawiszem **Ctrl** klawiatura ma zdefiniowane kilka przydanych znaków, np.: ©, §, °, ±, ÷.

PM – klawiatura polska z martwym klawiszem. Jako martwy klawisz zdefiniowano klawisz tyldy.

CF – klawiatura rosyjska z fonetycznym odwzorowaniem znaków (tzn. np. pod klawiszem **c** jest rosyjskie **с**, czyli **ц** graficznie).

CG – klawiatura rosyjska z graficznym odwzorowaniem znaków (tzn. pod klawiszem **c** jest rosyjskie **с**, czyli graficznie **с**).

GR – klawiatura grecka (pod klawiszem **a** jest litera **α**, pod **b** – **β**, itd.).

Użytkownik może w prosty sposób przełączać aktywne klawiatury, tj.: Ctrl + NUM1 włącza klawiaturę nr 1 (czyli znajdującą się na pierwszej pozycji listy załadowanych do pamięci klawiatur), Ctrl + NUM2 – drugą, itd.

Można też wybierać klawiatury poprzez wskaźnik aktywnej klawiatury, który znajdziemy na pasku stanu programu.



- Dwa pierwsze tryby (i) mają zastosowanie tylko w sytuacji, gdy aktywna jest ramka tekstowa, w przeciwnym wypadku, tryby te działają jak tryb komend, tzn. informacje z klawiatury przesyłane są do głównego okna programu.

O tym, jaki tryb pracy jest aktualnie włączony informuje nas wskaźnik na pasku stanu programu.




Dotknięcie myszką tego wskaźnika wywoła podświetlone menu, z którego wybieramy jeden z omówionych wyżej trybów. Ponadto tryby te możemy wybrać klawiszem **Insert** (przełączanie wstawianie/wymiana) lub kombinacją **Shift+Esc** (włączenie trybu programowania). Przejścia z trybu wstawiania na tryb komend i odwrotnie przy aktywnej ramce tekstowej dokonamy również klawiszem **Esc**. Włączenie trybu wstawiania przy aktywnej ramce tekstowej jest jednoznaczne z uaktywnieniem w niej kursora tekstowego.


Kody sterujące


Na podstawie korespondencji, którą otrzymujemy od początkujących użytkowników Kombi można stwierdzić, że tym elementem programu, który sprawia największą trudność jest pojęcie kodów sterujących. W poniższym artykule postaram się przybliżyć nieco to pojęcie.


W programie Kombi tekst jest traktowany jak strumień danych. Strumień ten składa się ze znaków (liter) przeplatanych informacjami sterującymi ich wyglądem na ekranie i wydruku. Analizator tekstu pobiera kolejne znaki (począwszy od pierwszego znaku w pierwszej ramce strumienia) i dokonuje ich interpretacji. Jeśli znak jest literą – to jest po prostu wyświetlany, w przeciwnym wypadku program dokonuje jego analizy i w zależności od jej wyniku podejmuje różne czynności. Znaki, które nie są literami, sterują więc procesem wyświetlania liter i stąd nazywam je kodami sterującymi. Można wyróżnić trzy typy kodów:

1. Kody „proste” – tj. takie, które nie zawierają żadnych powiązań z innymi danymi w programie. Są to:


 **Wymuszone przelanie.** Jeśli program natrafi na ten kod – porzuca wyświetlanie w bieżącej ramce i cały tekst od następnego znaku przenosi do kolejnej ramki w strumieniu. Kod ma zastosowanie np. na końcu rozdziału, jeśli chcemy aby rozdział następny rozpoczynał się od nowej ramki.


 **Potencjalny podział wyrazu.** Jeśli program natrafi na ten kod – zapamiętuje jego położenie. Jeśli potem okaże się, że wyraz nie mieści się w wierszu, program przeczuci wszystkie znaki znajdujące się za ostatnim zapamiętanym potencjalnym podziałem do nowego wiersza. Na etapie wyświetlania nie wykorzystane kody potencjalnego podziału wyrazu nie są wyświetlane, natomiast kody wykorzystane – są zastępowane znakiem dywizu. Opcjonalnie można włączyć tryb wyświetlania, w którym potencjalne podziały wyrazu są oznaczane na ekranie pionowymi kreskami.


 **Potencjalny podział ciągu znaków.** Kod działa podobnie do poprzedniego z jedną istotną różnicą. Otóż w miejscach wykorzystanego podziału nie jest wyświetlany dywiz. Kod ten może mieć zastosowanie do dzielenia ciągów znaków takich jak np. nazwy plików na dysku, adresy internetowe lub adresy poczty elektronicznej, np. <http://www.3n.com.pl>, można byłoby przenieść po znakach //, ale jednocześnie nie chcemy, aby po tych znakach program wyświetlił dywiz. Kombi zapewnia automatyczne dzielenie takich ciągów znaków zgodnie z listą **Podziały bez znaku po znakach**, którą edytujemy w konfiguracji modułu słownikowego. Dla opcji wyświetlania znaków potencjalnego podziału, te kody są oznaczane pionową przerywaną linią.


 **Logiczny koniec zdania.** Ten kod sterujący ma zastosowanie w stosunku do modułu słownikowego i z punktu widzenia tego modułu oznacza zwykłą kropkę kończącą zdanie. Natomiast z punktu widzenia modułu wyświetlającego – znak ten jest „przezroczysty” i nie jest wyświetlany. Kiedy powinniśmy użyć tego kodu? Np. po tytułach, które z reguły nie są kończone kropkami, ale logicznie stanowią zdanie. Wstawienie kodu pozwala


modułowi sprawdzającemu podzielić tekst na zdania i w konsekwencji poprawia jakość korekty.


 **Tabulator zwykły (lewostronny).** Kod działa jak zwykły tabulator w każdym edytorze tekstu, tzn. przesuwa wyraz w prawo do najbliższej pozycji tabulacji tak, aby lewa strona wyrazu pokrywała się z pozycją tabulacji.


 **Tabulator centrujący.** Chociaż program zapewnia standardową obsługę tabulatorów, polegającą na tym, że w tekście umieszczamy znaki tabulacji, a w wierszowniku (czyli na linii poziomej odmierzającej rozmieszczenie znaków w poziomie) ustalamy typ tabulatora, program dostarcza dodatkowych trzech tabulatorów rozszerzających możliwości równania tekstu. Znak tabulatora centrującego powoduje, że wyraz, przed którym ten znak się znajduje będzie centrowany względem najbliższej pozycji tabulacji, nawet jeśli pozycja ta nie jest centrująca. Umożliwia to różnorodne równanie wyrazów w kolejnych wierszach bazując na tej samej pozycji tabulacji w wierszowniku.

 **Tabulator prawostronny.** Ten tabulator działa analogicznie do tabulatora centrującego, ale wyrazy nim poprzedzone są równane do prawej krawędzi.


 **Koniec działania tabulatora.** Ten kod umożliwia uzyskanie dodatkowych efektów związanych z tabulacją i pracuje poprawnie w połączeniu z tabulatorem prawostronnym. Wyrazy (ciągi znaków), które mają być równane poprzedzamy tabulatorem prawostronnym. W tej sytuacji wyrazy równane są do prawej strony. Jeśli w wyrazach tych znajdzie się omawiany kod, to wyrównanie nastąpi do prawej krawędzi znaku, za którym ten kod stoi. Działanie to jest podobne do tabulatora dziesiętnego, z tym że znak tabulatora dziesiętnego jest stały dla wszystkich wyrazów równanych do konkretnej pozycji tabulacji, natomiast znak końca działania tabulatora pozwala wymusić w wyrazie (ciągu znaków) to miejsce, które zostanie uznane za pozycję tabulacji.

 **Przejdź do nowego wiersza (bez wyrównania wiersza poprzedzającego).** Kiedy program natrafi na ten kod, dalsza część tekstu zostanie przeniesiona do nowego wiersza. Między wierszem bieżącym, a następnym zostanie zastosowana standardowa odległość międzyliniowa. Wiersz bieżący nie będzie równany, nawet jeśli w akapicie obowiązujące równanie do obu marginesów.


 **Przejdź do nowego wiersza z wyrównaniem wiersza poprzedzającego.** Kod działa podobnie do poprzedniego, z tym że wiersz, w którym program natrafił na ten kod zostanie wyrównany zgodnie z parametrami wierszownika obowiązującego w tym miejscu.


 **Twarda spacja (o stałej szerokości).** Kod powoduje wstawienie w miejscu jego występowania przerwy (spacji) między sąsiednimi znakami. Spacja ta ma taką cechę, że w miejscu jej występowania program nie może „złamać” wiersza. Szerokość twardej


spacji jest cechą dokumentu i zadajemy ją w procentach względem spacji zwykłej. Twardą spację winniśmy stosować w połączeniach np. liczb i jednostek (np. 1 GB), dzięki czemu unikniemy przeniesienia jednostki do następnego wiersza.


 **Twarda spacja o zmiennej szerokości.** Ten kod jest „skrzyżowaniem” spacji zwykłej i twardej. Ze spacji twardej dziedziczy on cechę spajania sąsiadujących znaków. Ze spacji zwykłej natomiast, kod ten dziedziczy cechę zmiennej szerokości. Dzięki tym cechom możemy go wstawiać za jednoliterowymi wyrazami, co z jednej strony spowoduje niemożność pozostawiania ich na końcu wiersza, zaś z drugiej zapewni równomierne rozłożenie światła w wierszu, co nie jest możliwe przy wykorzystaniu zwykłej twardej spacji.


2. Kody „rozwijane”. Po ich napotkaniu, program wstawi w ich miejsce odpowiedni ciąg znaków. Są to:


 **Numer strony.** Po napotkaniu tego znaku, program wyliczy jaka liczba odpowiada numerowi strony, na której ten znak został napotkany i w jego miejsce wstawi ciąg znaków odpowiadający temu numerowi.

 **Numer strony plus jeden.** Ten kod działa podobnie do poprzedniego, z tym że do aktualnego numeru strony zostanie dodana wartość 1. Kod ten nie ma obecnie większego zastosowania i pozostawiono go dla kompatybilności z poprzednimi wersjami programu, w których służył on do numerowania stron w przypadku montażu dwóch stron na jednym arkuszu (np. 2 x A5 na A4).


 **Data.** Ten kod powoduje wyprowadzenie w miejscu jego wystąpienia aktualnej daty. Sposób wyświetlania daty (słownie, cyframi, itd.) może być definiowany globalnie dla dokumentu.

 **Czas.** Kod wyprowadza w miejscu jego wystąpienia aktualny czas systemowy. Format wyświetlanego czasu ustalamy globalnie dla dokumentu.

 **Numer strony, z której przelano tekst.** Ten kod działa podobnie do kodu numeru strony, z tym że zamiast numeru aktualnej strony, w tekst zostanie wstawiony numer strony, z której nastąpiło przelanie do ramki, w której znalazł się ten kod. Kod ma zastosowanie np. w stosunku do artykułów, które nie mieszczą się na jednej stronie i są kontynuowane na stronach kolejnych. Możemy wtedy na początku takiej kontynuacji umieścić tekst, np. „Dokończenie z strony x” i w miejsce znaku x wstawiamy omawiany kod. W tej sytuacji program wstawi w miejsce znaku x automatycznie właściwą wartość.

 **Numer strony, na którą nastąpi przelanie tekstu.** Kod działa podobnie do poprzedniego, ale umieszczamy go na końcu artykułu, np.: Dokończenie na stronie x.

3. Kody „interpretowane”. Te kody powodują, że program po ich napotkaniu ustala nowe parametry wyświetlania dalszych znaków.

 **Styl.** Kiedy podczas analizy kolejnych znaków tekstu program natrafi na ten kod, sprawdza zaszyty w nim indeks stanowiący pozycję na liście stylów ramki. Każda ramka ma swoją niezależną listę stylów. Ramki powiązane w strumień mają jedną, wspólną listę stylów. Po ustaleniu indeksu, program odczytuje z definicji stylu wszystkie informacje opisujące jak mają wyglądać znaki występujące po analizowanym kodzie. Większość parametrów

opisujących styl jest sprzężona z trójpoziwym przelącznikiem. Jego działanie omówimy na przykładzie kompresji liter. Jeśli w stylu, który analizujemy, przelącznik sprzężony z parametrem kompresji jest wyłączony, program od tego miejsca nie będzie kompresował liter (tzn. zastosuje kompresję 100-procentową). Jeśli przelącznik będzie włączony – program zastosuje kompresję zadaną w tym stylu. Jeśli przelącznik będzie w stanie nieokreślonym – program przyjmie kompresję zadaną w poprzednim stylu. Jeśli i w poprzednim stylu kompresja będzie nie określona – program cofnie się jeszcze o jeden styl, itd. aż do ewentualnego początku tekstu, gdzie zostanie przyjęta kompresja 100%.

Styl zawiera wszystkie informacje o wyglądzie znaków, nie zawiera natomiast informacji o sposobie ich rozmieszczania w ramce. Kodem, który ustala sposób rozmieszczenia znaków w akapicie jest wierszownik¹.

Powrót do poprzedniego stylu. Kiedy program trafi na ten kod sterujący, przywraca parametry stylu na takie, jakie obowiązywały przed ostatnim wystąpieniem parametru stylu. Analiza kolejnych wystąpień kodu stylu i powrotu stylu jest analogiczna do wielu znanych z innych programów rozwiązań opartych na tzw. tagach. Np. zupełnie podobnie pracuje html, gdzie oznacza zmianę stylu na wytłuszczenie, a oznacza powrót do stylu normalnego. Podobnie w plikach rtf: \b – to styl wytłuszczenia, a \b0 – powrót do stylu normalnego. Jeśli w Kombi chcemy zmienić tylko jeden parametr stylu, to ustalamy na „przejrzyste” te parametry, które nie mają ulec zmianie, a włączamy ten, który chcemy zmienić. Różnica między programem Kombi, a innymi rozwiązaniami obsługującymi się tagami polega na tym, że pojedynczy tag zmienia jeden parametr w stylu, natomiast kod zmiany stylu w Kombi zmienia dowolną liczbę parametrów zdefiniowanych w stylu (jeszcze raz podkreślę tu, że zmianie ulegają tylko te parametry, które mają zdefiniowany stan przelącznika związanej z danym parametrem). W pewnym sensie można więc powiedzieć, że kody sterujące są tagami. Nie zdecydowałem się jednak na takie ich nazwanie, ponieważ są one kodowane binarnie, a nie tekstowo tak jak zwykłe tagi. Kod powrotu do poprzedniego stylu był ważnym kodem w wersjach poniżej 7.x. **Uwaga!** Od wersji siódmej obsługa stylów opiera się na kodzie stylu, ale z racji kompatybilności wstecznej oraz niektórych specyficznych zastosowań – kod powrotu do poprzedniego stylu nadal jest obsługiwany.

Akapit. Kiedy program trafi na ten kod – postępuje podobnie jak w przypadku stylu, z tym, że znaleziony indeks kodu sterującego odnosi się do tablicy wierszowników. Podobnie jak w przypadku stylów, każda ramka zawiera własną listę wierszowników. Ramki połączone w strumień zawierają jedną – wspólną dla strumienia – listę wierszowników. Każdy akapit ma przypisany sobie wierszownik. Nie oznacza to jednak, że tekst składający się np. ze stu akapitów zawiera listę stu wierszowników. Te akapity, które mają takie samo formatowanie będą obsługiwane tym samym wierszownikiem.

Wcięcie akapitowe. Ten kod działa podobnie do kodu akapitu. Jedyna różnica polega na tym, że między akapit oznaczony tym kodem, a poprzedni nie zostanie dodana dodatkowa odległość międzyakapitowa.

Akapit ze zmienną odległością międzyakapitową. Ten kod działa identycznie do kodu akapitu. Ponadto wprowadza dodatkową możliwość związaną z pionowym równaniem tekstu w ramce. Otóż przy zadawaniu parametrów takiego równania tekstu możemy ustalić, aby do wyrównania kolumny program użył akapitów o zmiennej odległości międzyakapitowej. Wtedy całe nadmiarowe światło, które należy dodać, aby wyrównać tekst, zostanie rozłożone równomiernie na te akapity, które zostały oznaczone omawianym właśnie kodem. Jednym z zastosowań tego kodu jest np. przejście między podrozdziałami. Użycie tej możliwości spowoduje, że kolumna zostanie wyrównana w pionie kosztem światła między podrozdziałami, bez zmiany odległości między wierszami.

Ręczny kerning poziomy. Kiedy program napotka ten kod, odczytuje zawarty w nim parametr, który oznacza procentowe odsunięcie poziome dwóch sąsiadujących ze sobą znaków. Wartość 100% oznacza odsunięcie równe wysokości liter.

Przesunięcie pionowe. Kod działa podobnie do poprzedniego, ale dotyczy przesunięcia znaków w pionie. Przesunięcie równe 100%

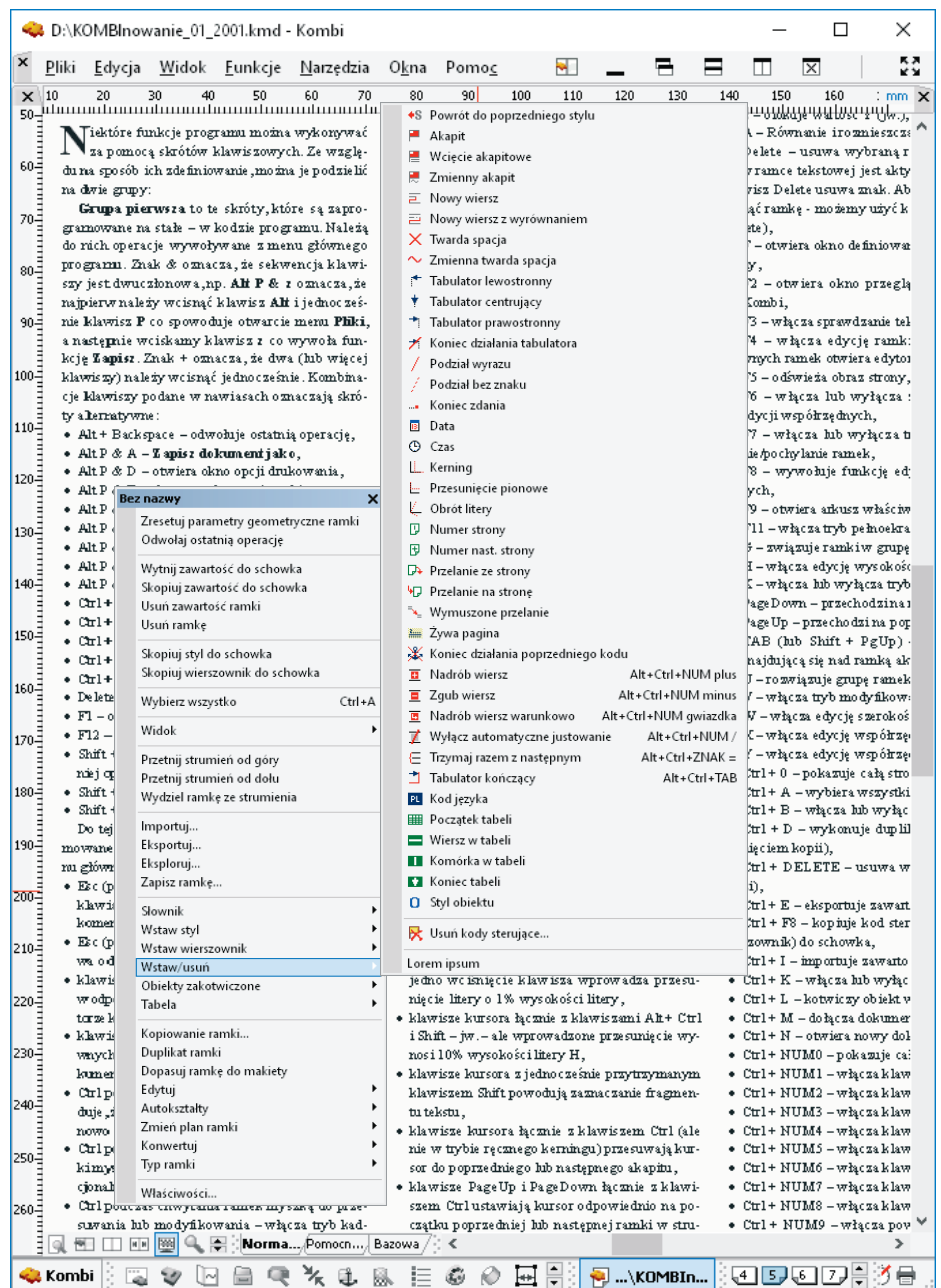
oznacza przesunięcie o wysokość litery. Możemy przesunąć znaki zarówno w górę, jak i w dół.

Obróć litery. Po napotkaniu tego znaku, wszystkie następne znaki będą obrócone o kąt zadany jako parametr w tym kodzie. Kąt zadajemy w stopniach w granicach od -90 do +90 stopni.

Obiekt zakotwiczony. Ten kod sterujący oznacza, że program ma wstawić w miejscu jego wystąpienia przypisany temu kodowi obiekt (ramkę, numerator, przypis, znacznik, łącze, itp.). Szczegółowy opis kotwiczenia obiektów znajdziemy w podręczniku użytkownika Kombi, tu zwrócę tylko uwagę na fakt, że ikony kodów sterujących pokazane w tym artykule zakotwiczone w tekście i będą one podążały za tekstem bez względu na zmiany wymiarów ramki oraz na zmianę ilości tekstu w ramce.

W wersji 8.x programu kody sterujące (poza stylem) wstawiamy na jeden z trzech sposobów:

1. Wykorzystując ikony na zakładce **Kody sterujące** w Autopilocie.
2. Wykorzystując ikony w palecie **Kody sterujące** w KombiKorze.
3. Za pomocą menu podręcznego ramki tekstowej.



¹ Szczegółowo pojęcia stylu i wierszownika omawiam w innym artykule tego numeru.

W standardowej konfiguracji programu, przy włączonej własnej obsłudze klawiatury, niektóre kody sterujące można wstawiać bezpośrednio z klawiatury. Są to następujące kody:

- Enter – akapit,
- Ctrl + Enter – wcięcie akapitowe,
- Shift + Enter – nowy wiersz bez wyrównania,
- Shift + Ctrl + Enter – nowy wiersz z wyrównaniem wiersza poprzedzającego,
- Shift + Ctrl + Spacja – twarda zmienna spacja,
- Shift + Spacja – twarda spacja,
- Ctrl + 6 – podział wyrazu,
- Ctrl + minus – podział wyrazu,
- Ctrl + 7 – podział bez znaku,
- Tab – tabulator lewostronny,
- Shift + Tab – tabulator prawostronny,
- Ctrl + Tab – tabulator centrujący,
- Ctrl + L – kotwiczenie obiektu.
- Alt + Ctrl + klawisze kursora i Alt + Ctrl + Shift + klawisze kursora wprowadzają ręczny kerning² w odpowiednim kierunku (bez klawisza Shift – 1%, z klawiszem Shift – 10% wysokości znaku).

Naturalną konsekwencją przyjęcia opisanego wyżej sposobu traktowania tekstu jako strumienia danych, jest zaakceptowanie faktu, że kody sterujące w tekście zachowują się jak zwykłe znaki. Tak więc przesuwanie kursora „wyczuwamy” w tekście nawet te kody, które są niewidoczne i pomimo ich niewidoczności można na nich wykonywać operacje za pomocą klawiatury. Można więc względem nich stosować wszystkie zabiegi takie jak dla zwykłego tekstu, a więc usuwanie, kopiowanie i wklejanie za pomocą operacji na schowku, a także wyszukiwanie i zamiana za pomocą standardowego okna Szukaj/Wymień. Trzeba również nadmienić, że owa niewidoczność niektórych kodów sterujących występuje tylko w trybie graficznym, tzn. kiedy oglądamy tekst w ramach na stronie. Zawsze możemy otworzyć edytor tekstu (standardowo skrót **F4**) przy aktywnej ramce tekstowej i w nim wszystkie kody sterujące są markowane odpowiednimi symbolami (zgodnymi z ikonami przypisanymi tym kodom).

W trybie graficznym natomiast, program dostarcza jeszcze jednej techniki operowania na kodach sterujących. Można mianowicie włączyć tryb pokazujący kodów sterujących w ramce tekstowej (menu podręczne ramki tekstowej – podmenu **Widok - Pokaż kody sterujące**). W trybie tym kody sterujące markowane są na ekranie w postaci chorągiewek zawierających symbol i (lub) opis kodu. Można kliknąć prawym przyciskiem myszki w taką chorągiewkę i z otwartego w ten sposób menu wybrać pozycję **Opcje zobrazowania kodów**. W otwartym w ten sposób oknie ustalamy, które kody mają być widoczne oraz sposób ich pokazywania. Wspomniane wyżej menu podręczne zawiera również dodatkowe funkcje ułatwiające operowanie na kodach sterujących, w tym pozycję **Właściwości**, która otworzy okno właściwości danego kodu sterującego. Pozycja ta jest domyślna w menu co oznacza, że ten sam efekt uzyskamy klikając dwukrotnie myszką w wybrany kod.

Jak wyżej wspomniano, opisane sposoby wprowadzania kodów sterujących nie odnoszą się wprost do kodu stylu. Ten kod, ze względu na swoją specyfikę jest obsługiwany w sposób szczególny. Jeśli zaznaczymy w ramce fragment tekstu

² Patrz artykuł: **Tryby pracy klawiatury**.

i zmienimy jakikolwiek parametr decydujący o wyglądzie tekstu (np. zmienimy kolor w palecie atrybutów lub na liście krojów wskażemy inny krój, czy też zmienimy wielkość znaków), to program wstawi sam w odpowiednie miejsca w tekście dwa kody sterujące, tj. w miejscu początku zaznaczonego fragmentu tekstu zostanie wstawiony kod stylu opisujący nowy wygląd znaków oraz w miejscu końca zaznaczonego fragmentu – kod odpowiadający stylowi obowiązującemu w tym

miejscu. Tak więc użytkownik (zwłaszcza początkujący) nie ma potrzeby wprowadzania kodu stylu w sposób jawny, choć oczywiście taka możliwość istnieje.

Ponieważ właściwe stosowanie kodów stylu i wierszownika jest bardzo istotne dla pracy programu poświęćmy temu zagadnieniu osobny artykuł w tym numerze, do którego odsyłam zainteresowane osoby.



Style i wierszowniki

W wielu listach od osób próbujących poznać program pojawia się pytanie o to, czemu wprowadzono styl i wierszownik, a nie jeden tzw. styl akapitowy. W poniższym artykule postaram się nieco wyjaśnić ten problem.

Zacznijmy od zdefiniowania pojęć stylu¹ i wierszownika² w programie Kombi. Przez styl rozumiem wszystkie te cechy znaków, które nie zależą od ich położenia na stronie. Innym dobrym określeniem stylu byłoby więc słowo **wygląd**. A więc styl definiuje **wygląd** znaku.

Wierszownik natomiast zawiera te cechy, które nie wpływają na wygląd, ale na **rozmieszczenie** znaków w ramce czy na stronie.

Tak więc, jeśli zaznaczę kod³ stylu (czyli wyglądu) znaków w akapicie który jest np. centrowany i skopiuję go do schowka, a następnie wkleję ten kod w akapit, który jest np. równany w chorągiewkę, to zmienię wygląd znaków pozostawiając ich układ nie zmieniony. Podobnie jeżeli pobiorę dane o układzie znaków w jakimś akapicie i przeniosę te dane na inny akapit – to zmienię układ znaków w tym drugim akapicie nie zmieniając jednocześnie wyglądu znaków w nim zawartych.

Rozwiązanie to pozwala opisać stronę mniejszą liczbą potrzebnych do tego kodów sterujących. Jeśli bowiem zdefiniuję np. pięć sposobów ułożenia tekstu w pracy (czyli pięć wierszowników) i pięć różnych wyglądków znaków (czyli stylów), to mam w sumie dwadzieścia pięć możliwości ich kombinacji zdefiniowanych za pomocą dziesięciu kodów sterujących. W rozwiązaniu nie rozdzielałym stylu od wierszownika, musiałbym w analogicznej sytuacji zdefiniować dwadzieścia pięć „stylów akapitowych”. Oczywiście w praktyce nie wszystkie one byłyby potrzebne, ale wskazuję tu na możliwość łączenia różnych wierszowników z różnymi stylami.

Czy wobec tego „styl akapitowy” nie ma zalet? No cóż, nie ma róży bez kolców. Wstawiając jeden styl akapitowy ustalam zarówno wygląd, jak i układ znaków w akapicie. W Kombi muszę wybrać dwa parametry dla akapitu, tj. obowiązujący w nim wierszownik i styl. Jednak czynność tę wykonujemy raz dla ramki (czy nawet dla całego strumienia), najlepiej dla pierwszego akapitu ramki.

¹ Styl – „rylec do pisania na tabliczkach powleczonej wojskiem, używany w starożytności” – **Maly Słownik J. Polskiego** pod red. S. Skorupki.

² Wierszownik – „przrząd zecerski służący do ręcznego układania czcionek drukarskich w wiersze” – j.w.

³ Pojęcie **kodu sterującego** wyjaśniłem w innym artykule tego numeru.

Poza wykazanymi wyżej różnicami w przedmiocie, który jest definiowany stylem i wierszownikiem, kody te różnią się również zakresem działania. Parametry ustalone w wierszowniku obowiązują w całym akapicie, do którego przypisano dany wierszownik. Kod zmiany stylu obowiązuje natomiast od momentu jego wystąpienia, do następnej zmiany. Bez znaczenia jest przy tym granica akapitu. Tzn. można zmienić styl w jednym akapicie, a przywrócić styl poprzedni dopiero w akapicie następnym.

Kolejnym argumentem przemawiającym za rozdzieleniem parametrów na dwie grupy jest uproszczenie definiowania każdego z kodów poprzez zmniejszenie liczby parametrów wymaganych do ich definiowania. Pełna definicja stylu zawiera 15 parametrów (poza kolorami), a pełna definicja wierszownika – 12. Połączenie ich w jeden kod wymagałoby od użytkownika zadeklarowania wszystkich 27 parametrów.

Sprawa następną – to możliwość oddzielenia w czasie etapu wprowadzania w tekst kodów sterujących wyglądem liter, od etapu polegającego na układaniu tekstu w łamy. Etap pierwszy – (tzn. wprowadzanie wyróżnień w tekst) może odbywać się wstępnie już w czasie wpisywania tekstu. Tzn. wprowadzając tekst, operator może oznaczyć te fragmenty tekstu, które chce w jakiś sposób wyróżnić (np. wytłuszczyć lub użyć kursywy, itp.). Trudno natomiast wymagać, aby na tym etapie wprowadzać dane dotyczące układu tekstu w ramce, bowiem zależą one od przyjętych później parametrów składu.

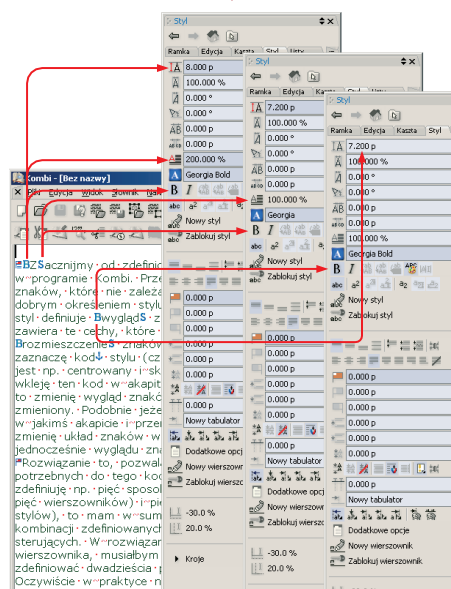
To wstępne wprowadzenie wyróżnień jest możliwe, ponieważ styl jest kodem sterującym i można go traktować jak tekst. Można więc ustalić z osobą wpisującą tekst pewne umowne oznaczenia, np. – może oznaczać wytłuszczenie, a <p> powrót do stylu poprzedniego. Potem, na etapie składu wystarczy wymienić owe oznaczenia na właściwe kody stylu.

Po ustaleniu, czym jest w Kombi styl, a czym wierszownik, zobaczymy jak w programie posługiwać się efektywnie tymi kodami. Rozpatrzmy najpierw rozwiązanie najprostsze. Mamy w ramce tekst i chcemy zmienić wygląd (nie układ) znaków w tej ramce. Jak to zrobimy? Założmy, że parametrem, który chcemy zmienić będzie kolor. A więc w palecie atrybutów wypełnień wybieramy nowy kolor (przez wskazanie go na liście kolorów). Program winien natychmiast odpowiedzieć zmianą koloru liter w ramce. Jaki



będzie zakres wprowadzonych zmian? Algorytm jest następujący. Program odszukuje w tekście miejsce, w którym znajduje się kursor i podąża od tego miejsca w lewo do momentu napotkania kodu stylu. Po napotkaniu go, program sprawdza, czy ten styl ma włączoną aktywność koloru, czy nie. Jeśli tak, to zmienia kolor zapamiętany w tym stylu na nowy kolor wskazany w palecie. Jeśli nie (tzn. napotkany styl, nie ma włączonej aktywności koloru), program analizuje dalej kolejne znaki w lewo, aż do napotkania następnego kodu stylu i tak, aż do ewentualnie pierwszego stylu w ramce.

Tak więc zmieniony parametr będzie obowiązywał od najbliższego kodu stylu w lewo od kursora, do najbliższego kodu stylu lub kodu powrotu do stylu poprzedniego w prawo od kursora. Jeśli zmienimy w ten sposób jakiś parametr stylu, a dalsze kody stylu w prawo od kursora nie będą miały tego parametru włączonego jako aktywny, to zmiana ta przeniesie się na wszystkie następne style w ramce. Zmiana nie przeniesie się, jeśli któryś z napotkanych kodów stylu w prawo od kursora będzie miał ten zmieniany parametr ustalony jako aktywny. Rozpatrzymy to na przykładzie np. wytłuszczenia. Oto początkowy fragment tego artykułu pokazany w edytorze tekstu i parametry odpowiadające trzem pierwszym stylom w ramce:



Pierwszy kod stylu w ramce (B) oznacza styl dla inicjału. Wielkość liter dla tego stylu jest równa 8 p, a przycisk przy tej wartości jest włączony, co oznacza, że program przyjmie w tym miejscu tę wartość. Wielkość inicjału jest dla tego stylu ustalona na 200% (czyli dwa wiersze) i przycisk przy tej wartości jest również włączony, co oznacza aktywność tego atrybutu. Ponadto styl ma włączony parametr wytłuszczenie (ikona B). Styl drugi ma wielkość ustawioną na 7.2 p, a przyciski przy inicjale i przy wytłuszczeniu są wyłączone. Spójrzmy jeszcze na styl trzeci (B). Ten styl ma włączony przycisk wytłuszczenia, ale przycisk przy parametrze ustalającym wielkość znaków jest w stanie niezdefiniowanym.

Jakie konsekwencje wynikają z takiego zdefiniowania parametrów? Jeśli w drugim stylu zmienimy wielkość np. z 7.2 p na 10 p, to trzeci styl dostosuje się do drugiego i również przyjmie 10 p (ponieważ przycisk przy wielkości liter nie jest włączony). Gdyby przycisk ten był włączony, to trzeci styl zachowałby swoją wielkość 7.2 p bez względu na zmiany wielkości w stylu drugim. Tak więc wstawiając styl trzeci w dowolne miejsce w ramce powodujemy

w tym miejscu zmianę atrybutu na **wytłuszczony** pozostawiając jednocześnie pozostałe parametry nie zmienione.

Tu przy okazji mała dygresja na temat wytłuszczenia i kursywy. Chociaż wiele programów traktuje te parametry jak atrybuty danego kroju, tak naprawdę, to aby uzyskać właściwy skład należy użyć innych odmian tego samego kroju. Co prawda system operacyjny umożliwia tzw. syntezę odmian, tzn. może z odmiany normalnej wygenerować np. odmianę pochyłą, ale w Kombi zrezygnowano z takiego rozwiązania, gdyż jest ono nieprofesjonalne. W Kombi, aby użyć wytłuszczenia lub pochylenia należy użyć po prostu innego kroju. Stąd też dostępny w menu stylów parametr włączający pochylenie znaków, należy traktować jako możliwość uzyskiwania efektów specjalnych, nie zaś jako włączenie kursywy.

Opisane wyżej operacje na stylach odnoszą się do takiej sytuacji, w której mamy już w ramce zaznaczone style i chcemy je tylko modyfikować. Jak natomiast wprowadzamy w tekst nowe style? Postępujemy podobnie, z tym, że zaznaczamy w ramce fragment tekstu i dopiero wtedy zmieniamy parametry. Spowoduje to wstawienie w miejscu początku zaznaczenia nowego stylu, a w miejscu końca zaznaczenia stylu takiego, który odpowiada parametrom tekstu w tym miejscu.

Podsumujmy więc: jeśli zaznaczymy fragment tekstu i zmienimy parametry stylu – program ustali nowe parametry tekstu w zaznaczonym fragmencie. Jeśli nie zaznaczymy fragmentu tekstu – program zmieni parametry stylu obowiązującego aktualnie w miejscu, w którym znajduje się kursor (a więc nie zostanie wstawiony nowy kod stylu).

Posługując się omówioną wyżej techniką można w zasadzie składać dowolne prace, ale jest to technika przydatna bardziej do małych prac typu: wizytówka, reklamówka czy etykieta. W składzie dzielowym przydatna jest inna technika, również dostępna w Kombi, a oparta na listach stylów i wierszowników.

Każda ramka (strumień) w dokumencie Kombi ma tzw. listę stylów i osobną listę wierszowników. Są to listy **lokalne**. Ponadto dwie identyczne listy istnieją na poziomie dokumentu (są to listy **globalne**). Kiedy program tworzy nowy styl lub wierszownik umieszcza go na liście lokalnej (to się dzieje automatycznie bez względu na to czy użytkownik chce korzystać z tych list, czy też nie). Użytkownik może wybrać styl lub wierszownik z listy lokalnej i wkleić go w dowolne inne miejsce. Operacja ta jest wykonywana, kiedy klikniemy w wybraną pozycję na liście. W efekcie takiego działania mamy w kilku miejscach odwołanie do tej samej pozycji na liście i w konsekwencji zmieniając parametry tej pozycji zmieniamy parametry składu w kilku miejscach jednocześnie.

W przeciwieństwie do listy lokalnej – kody na liście globalnej nie są tworzone automatycznie. Tu użytkownik musi jawnie stworzyć styl lub wierszownik i ustalić jego atrybuty. Kliknięcie w pozycję na liście globalnej powoduje utworzenie kopii kodu na liście lokalnej, ale kopia ta jest związana z kodem źródłowym, dzięki czemu zmiany parametrów kodów na liście globalnej przenoszą się na odpowiednie kopie tych kodów w ramkach.

Przy korzystaniu z listy stylów obowiązują następujące zasady:

- Jeśli w tekście jest zaznaczony fragment tekstu – program wstawi w miejscu początku zaznaczenia nowy styl, a w miejscu końca zaznaczenia – kod odpowiadający parametrami stylowi, który w tym

miejscu obowiązuje. Jak wynika z powyższego program może być zmuszony do utworzenia zupełnie nowego (do tej pory nie istniejącego stylu), aby zapewnić nie zmienione parametry stylu za zaznaczonym fragmentem.

- Jeśli wstawiając styl z listy przytrzymamy wciśnięty klawisz **Ctrl** – program przed wstawieniem stylu zaznaczy akapit, w którym znajduje się kursor.
- Sposób oddziaływania wprowadzanego w tekst stylu z listy z tekstem w ramce zależy od atrybutu stylu **Łącz parametry**. Jeśli dla danego stylu ten parametr jest włączony i w tekście jest zaznaczony fragment, to wklejenie stylu spowoduje zmiany w zaznaczonym fragmencie tekstu tylko tych parametrów, które są zadeklarowane jako aktywne we wprowadzanym stylu. Jeśli wspomniany wcześniej parametr nie jest włączony – style znajdujące się wewnątrz zaznaczonego fragmentu zostaną usunięte. Przytrzymanie klawisza **Shift** w tej sytuacji zmienia chwilowo stan opcji **Łącz parametry** na przeciwny.
- Jeśli podczas wklejania stylu z listy w tekście nie zaznaczono fragmentu – ani opcja **Łącz parametry**, ani klawisz **Shift** nie mają wpływu na akcję. Natomiast klawisz **Ctrl** spowoduje zaznaczenie akapitu, w którym znajduje się kursor.
- Jeśli w ramce nie ma zaznaczenia – jest wstawiany w miejscu kursora wybrany z listy kod stylu.
- W przypadku listy wierszowników przy wstawianiu wierszownika w tekst obowiązują wymienione niżej zasady:
- Jeśli w tekście nie zaznaczono fragmentu – uaktualniane są dane w tym akapicie, w którym stoi kursor.
- Jeśli w tekście jest zaznaczony fragment – uaktualniane są dane we wszystkich akapitach znajdujących się wewnątrz zaznaczenia z początkowym i końcowym włącznie.
- Klawisz **Shift** działa identycznie jak dla stylów (tzn. zmienia chwilowo stan opcji **Łącz parametry**) na przeciwny.
- W przypadku wstawiania wierszowników użyć klawisza **Ctrl** nie ma znaczenia.

Parametry stylu lub wierszownika, który został automatycznie lub ręcznie utworzony i dodany do listy można zmienić poprzez arkusz właściwości stylu lub wierszownika. Arkusz ten otworzymy funkcją **Właściwości** w menu podręcznym listy.

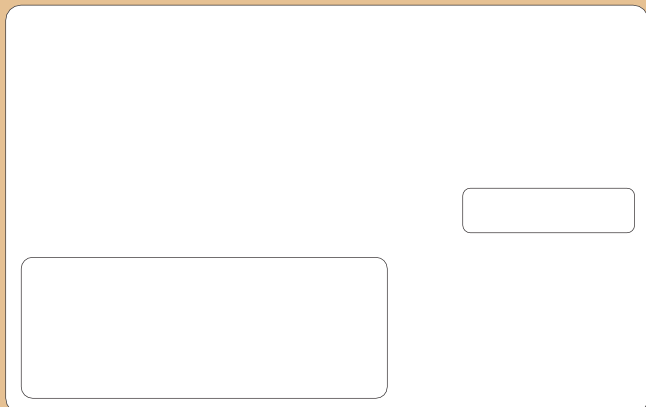
Oczywiście w ten sposób zmienione parametry kodu przeniosą się na wszystkie miejsca w ramce (w przypadku listy lokalnej) lub w całym dokumencie (w przypadku listy globalnej), w których edytowany kod się znajduje.

Należy tu jeszcze zwrócić uwagę na dwie funkcje dostępne w menu podręcznym listy lokalnej. Mam tu na myśli funkcje: **Podłącz do stylu globalnego** i **Odlącz od stylu globalnego**.

Jaki jest sens użycia tych funkcji? Można np. zaimportować tekst w formacie * rtf i po dokonaniu wstępnej obróbki podłączyć kody lokalne (z zaimportowanego pliku) do globalnych kodów zdefiniowanych na poziomie dokumentu. Możliwa jest też operacja odwrotna, tzn. styl, który został wprowadzony z listy globalnej można odłączyć od tej listy przez co stanie się on kodem lokalnym i będzie go można zmodyfikować bez ryzyka wprowadzenia zmian w innych miejscach, w których styl ten występuje.

Co i jak ?

W kolejnych numerach poradnika będę pokazywał co i jak można zrobić w Kombi. Dziś – karta stałego klienta.



1. Zaczynamy od zamarkowania obrysów pracy. Aby uzyskać zaokrąglone narożniki, użyłem predefiniowanego kształtu i zadałem wymagany promień zaokrąglenia. Zaznaczyłem też obszary, które będą poddane rozjaśnieniu. Ramki z kształtem konwertuję na wektorowe i sklejam w jedną ścieżkę. (Funkcja **konwertuj na ramkę wektorową** konwertuje kształty na wektory, zaś w przypadku użycia jej względem kilku ramek wektorowych – spowoduje ich sklejanie).

2. Przygotowuję zdjęcie z aparatu cyfrowego lub skanera. Dobieram właściwe wymiary i umieszczam w ramce z bitmapą.

3. Ramkę wektorową umieszczam nad ramką z bitmapą. Aby wymiary obu ramek pokrywały się – korzystam z wymiarowania za pomocą palety współrzędnych lub z funkcji **Równanie/Rozmieszczanie ramek**.

4. Ramkę z wektorem przenoszę do własnego schowka programu.

5. Uaktywniam ramkę z bitmapą i wchodzę do edycji ścieżki w bitmapie. Następnie wklejam ścieżkę ze schowka. Wklejona ścieżka wektorowa staje się ścieżką odcinania w bitmapie.

6. Obrabiam bitmapę za pomocą filtru **Matówka** i rozmywam przejścia między obszarem rozjaśnionym i nie rozjaśnionym za pomocą narzędzia **Rozjaśniacz** w edytorze grafiki rastrowej.

KOMBI

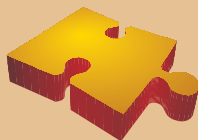
7. Napis KOMBI umieszczam w ramce tekstowej i zadaję atrybuty – wypełnienie tonalne dla wnętrza i obrysu cienia. Obrys pierwszego planu jest cieniem linią koloru szarego.

v. 8.0

N 123456

© 3N 1996-2007

8. Proste napisy umieszczam w ramach tekstowych utworzonych w odpowiednich miejscach na karcie.



9. Puzzle został narysowany jako wektor i obrobiony w **Projektorze 3D**. Następnie konwertuję go na ramkę z bitmapą, umieszczam na karcie i zadaję przezroczystość.



10. Zakładam nową ramkę wektorową, generuję do niej kod kreskowy i umieszczam go na karcie.

Karta stałego klienta nr

11. W ramce tekstowej umieszczam napis. Ramkę kopiuję i dodaję rozmycie krawędzi. Tak otrzymaną ramkę konwertuję na ramkę z bitmapą, nakładam na kartę i ustaliam przezroczystość bitmapy. Na pierwszy plan nakładam napis w ramce tekstowej.