Jarosław Kuchta

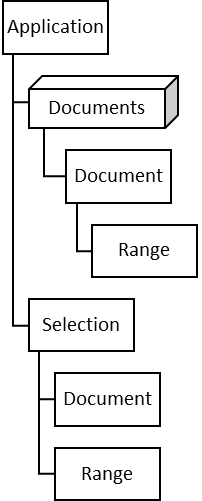
Model dokumentu Interop Word

data utworzenia: 27.10.2023

data ostatniej modyfikacji: 22.12.2023

# Wprowadzenie

Deweloperzy aplikacji .NET współpracujących z programem Microsoft Word mogą korzystać z pakietu narzędzi VSTO i kilku bibliotek połączonych wspólną przestrzenią nazw Microsoft.Office.Interop.Word. Przestrzeń ta zawiera definicje obiektów COM (Common Object Model) tworzących model aplikacji i dokumentu Word. Obiekty COM są udostępniane przez program Word dla aplikacji przez wywołania swoich publicznych interfejsów. Tworzą one hierarchię w formie drzewa, którego korzeniem jest obiekt Application (rys. 1).



Rys. . Najwyższy poziom struktury interfejsów przestrzeni nazw Microsoft.Office.Interop.Word

Programista korzystający z VSTO ma wybór: może tworzyć dodatki do aplikacji Word działające na poziomie aplikacji lub działające na poziomie dokumentu. W pierwszym przypadku narzędzia VSTO tworzą obiekt klasy Application, którego właściwości i metody wywoływane w programie dają dostęp do pozostałych obiektów dokumentu. W drugim – programista dysponuje obiektem ThisDocument i może bezpośrednio korzystać z właściwości i metod tego dokumentu albo (przez jego właściwość Application) uzyskać dostęp do właściwości i metod aplikacji Word.

Drugim po Application, najważniejszym obiektem modelu jest obiekt klasy Document. Reprezentuje on sam dokument i jego zawartość. Dokumenty tworzone w aplikacji Word są dodawane do kolekcji Documents obiektu Application. Programista może operować na dokumencie aktualnie wybranym w aplikacji Word korzystając z właściwości ActiveDocument obiektu Application lub na dowolnym dokumencie wybranym z kolekcji Documents poprzez numer lub nazwę dokumentu.

Kolejnym, bardzo ważnym elementem struktury dokumentu jest obiekt klasy Range (zakres). Reprezentuje on pewien fragment dokumentu określony przez pozycje znakowe początku i końca zakresu i umożliwia dostęp do takich swoich elementów jak akapity (z podziałem na zdania i wyrazy), tabele (w tym wiersze, komórki), ramki, pola, przypisy etc. Klasa dokumentu ma metodę Range tworzącą zakres obejmujący podane pozycje początku i końca (oczywiście można ustawić zakres na całą treść dokumentu). Każdy z wybranych elementów dokumentu (np. akapit czy tabela) ma właściwość Range podającą zakres tego elementu. W ten sposób programista może działać na wielu zakresach w dokumencie, chociaż działania te nie będą od razu prezentowane w aplikacji Worda.

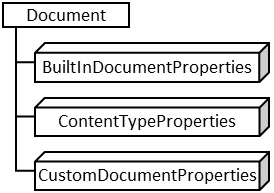
Nieco inaczej jest z obiektem klasy Selection. W aplikacji jest tylko jeden obiekt tej klasy reprezentujący zakres aktualnie zaznaczony w aktywnym dokumencie. Jeśli nic nie jest zaznaczone, to pozycja początku i jednocześnie końca tego zakresu jest pozycją, w której ulokowany jest kursor tekstowy. Zmiany dokonane przez obiekt klasy Selection są na bieżąco prezentowane w aplikacji Worda (co z jednej strony umożliwia wizualną kontrolę wprowadzanych zmian, ale z drugiej strony powoduje wolniejsze wykonywanie instrukcji).

# Dokument i jego struktura

Dokument w VSTO jest reprezentowany przez obiekt implementujący interfejs Document, który łączy w sobie podstawowy interfejs (ang. primary interface)[[1]](#footnote-1) \_Document (definiujący właściwości i metody dokumentu) oraz interfejs DocumentEvents2\_Event (umożliwiający podłączenie aplikacji .NET do uchwytów zdarzeń aplikacji Word).

## Właściwości dokumentu, ustawienia

Pojęcie właściwości dokumentu (ang. document properties) trzeba odróżnić od pojęcia właściwości klasy dokumentu (ang. document class properties) albo też właściwości obiektowych dokumentu (ang. document object properties). Pod tym pierwszym pojęciem rozumie się *metadane*, czyli informacje o dokumencie, które opisują jego zawartość. Wyróżnia się trzy grupy właściwości dokumentu (rys. 2):



Rys. . Właściwości dokumentu

* właściwości wbudowane,
* właściwości typu zawartości,
* dowolne właściwości dokumentu.

Właściwości wbudowane dokumentu są reprezentowane przez kolekcję BuiltInDocumentProperties. Udostępnia ona takie metadane, jak:

* tytuł dokumentu,
* datę utworzenia,
* oznaczenie autora, który utworzył dokument,
* datę ostatniej modyfikacji,
* oznaczenie autora, który ostatnio zmodyfikował dokument, etc.

Większość tych metadanych została zdefiniowana w standardzie Dublin Core[[2]](#footnote-2). który dotyczy dokumentów różnych typów. Właściwości specyficzne dla dokumentu obsługiwanego przez program Word są zdefiniowane w kolekcji ContentTypeProperties. Są to takie dane jak:

* nazwa firmy, do której należy dokument,
* oznaczenie kierownika projektu, do którego należy dokument,
* nazwa i nr wersji aplikacji, która utworzyła dokument,

ale także:

* liczba stron dokumentu,
* liczba akapitów,
* liczba linii (wierszy tekstu),
* liczba słów,
* liczba znaków (ze spacjami i bez spacji).

Właściwości z ostatniej podgrupy są obliczane na bieżąco i odzwierciedlają aktualny stan dokumentu.

Trzecia grupa właściwości zapisanych w kolekcji CustomDocumentProperties obejmuje dowolne inne metadane, które nie zostały uwzględnione w poprzednich dwóch grupach i mogą być swobodnie definiowane przez autorów dokumentu.

Inne informacje o dokumencie są udostępniane przez właściwości obiektu Document. Można powiedzieć, że określają one aktualny stan dokumentu. Są to m.in.:

* typ dokumentu (szablon albo dokument),
* nazwa i ścieżka pliku dokumentu,
* kodowanie znaków,
* typ ochrony dokumentu.

Podobnie przez właściwości obiektu Document ustalane są ustawienia (ang. settings). Są to w większości wartości typu logicznego lub liczbowego, które określają sposób operowania aplikacji Worda na danym dokumencie. Jako przykłady można podać:

* zezwolenie na dzielenie wyrazów pisanych wielkimi literami,
* określenia odstępów linii siatki w pionie i w poziomie,
* określenie czy znaczniki poprawek są drukowane razem z dokumentem.

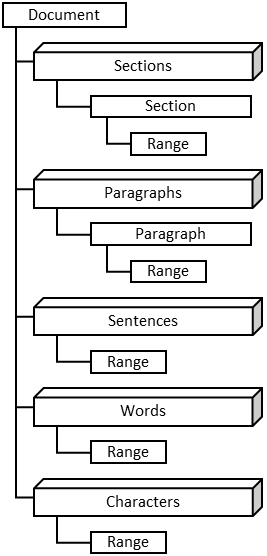
Dostępność poszczególnych ustawień jest silnie zależna od wersji aplikacji Worda, która obsługuje danych dokument.

## Identyfikacja dokumentu i jego wersji

Nazwa dokumentu, nazwa pliku, a nawet ścieżka pliku nie są wystarczające do jednoznacznej identyfikacji dokumentu. Obiekt Document ma właściwość DocID, która jest jednak oznaczona jako „do użytku wewnętrznego”.

Innym problemem jest oznaczenie wersji dokumentu. Kolekcja BuiltInDocumentProperties przechowuje metawłaściwość (ang. metaproperty) Version (typu tekstowego), która może być dowolnie używana przez autorów dokumentu. Oprócz tego aplikacja Word automatycznie przypisuje do dokumentu losową liczbę CurrentRsid, która służy do oznaczania zmian w dokumencie. Dokument ma zdefiniowaną kolekcję Revisions, która działa wtedy, gdy śledzenie zmian w dokumencie jest włączone (właściwość TrackRevisions jest ustawiona na true). Dokument ma też zdefiniowaną kolekcję Versions, która powinna zwracać informację o wersjach dokumentu, ale to działa tylko wtedy, gdy dokument jest zapisywany na dysku OneDrive lub w bibliotece SharePoint.

## Zawartość tekstowa



Rys. . Kolekcje elementów tekstowych

Tekst dokumentu jest udostępniany przez szereg kolekcji (rys. 3):

* Sections – reprezentuje sekcje dokumentu,
* Paragraphs – reprezentuje akapity,
* Sentences – reprezentuje zdania,
* Words – reprezentuje wyrazy,
* Characters – reprezentuje znaki.

Sekcji nie należy utożsamiać z rozdziałami i podrozdziałami. Sekcja to pewien zakres dokumentu, który ma wspólne właściwości, takie jak:

* rozmiar papieru,
* marginesy stron,
* liczba kolumn,
* nagłówki i stopki.

Jedna sekcja może obejmować wiele stron, ale też na jednej stronie może znajdować się wiele sekcji, np. różniących się liczbą kolumn (rys. 4).

Sekcja

1-kolumnowa

Sekcja

2-kolumnowa

*Rys. 4 Dwie sekcje na jednej stronie*

Sekcja ma właściwość Range reprezentującą zakres, a przez ten zakres daje programiście dostęp do swoich akapitów, zdań, wyrazów i znaków.

Drugim z kolei (i właściwie głównym) elementem tekstowym jest akapit (Paragraph). Akapit jest ciągiem tekstowym zakończonym znacznikiem końca akapitu (którego nie można mylić ze znacznikiem podziału wiersza).

Akapit ma swoje właściwości, takie jak:

* wyrównanie tekstu,
* wcięcie z lewej, z prawej,
* odstępy międzyliniowe,
* odstępy przed i po akapicie.

Zbiór właściwości formatujących akapit reprezentuje właściwość Format typu ParagraphFormat. Dzięki temu właściwości formatujące akapit mogą być traktowane jak pojedynczy obiekt, duplikowane i przypisywane do innych akapitów.

Zakres akapitu udostępnia jego tekst, który może być odczytany jako łańcuch znaków, a może też być podzielony na zdania, słowa i znaki.

Zdanie jest ciągiem znaków w akapicie zakończonym kropką znakiem zapytania lub wykrzyknikiem. Algorytm podziału akapitu na zdania nie odróżnia kropek po skrótowcach, dlatego efekt może nie być zgodny z podziałem gramatycznym.

Zdanie nie ma osobnej klasy elementów. Kolekcja Sentences zwraca zakresy (Range).

Wyraz to ciąg liter, cyfr albo innych symboli (z pominięciem spacji). Wyraz również nie ma osobnej klasy elementów. Kolekcja Words, tak jak kolekcja Sentences, zwraca obiekty typu Range.

Najdrobniejszym elementem tekstowym jest znak. Również kolekcja Characters zwraca elementy typu Range.

## Formatowanie tekstu i akapitu

Zakres (Range) ma właściwość Font określającą formatowanie znaków oraz właściwość ParagraphFormat reprezentującą formatowanie akapitów.

Obiekt klasy Font ma wiele właściwości dotyczących czcionki używanej do prezentacji danego fragmentu tekstu. Są to m.in.:

* nazwa czcionki (Name, NameAscii, NameBi, NameFarEast, NameOther),
* rozmiar czcionki (Size, SizeBi),
* wytłuszczenie (Bold, BoldBi),
* kursywa (Italic, ItalicBi),

Tu trzeba zauważyć, że aplikacja Word umożliwia stosowanie w tym samym zakresie niezależnych czcionek dla różnych rodzajów pisma: opartych o alfabet łaciński, pisma dwukierunkowego (stosowanego na Bliskim Wschodzie), pisma dalekowschodniego (chińskiego, koreańskiego, japońskiego) i innych. Umożliwia również stosowanie różnych wielkości znaków, wytłuszczenia i kursywy dla pisma jedno- i dwukierunkowego.

Ponadto obiekt klasy Font ma właściwości określające;

* podkreślenie (Underline),
* przekreślenie (StrikeThrough),
* przekreślenie podwójne (DoubleStrikeThrough),
* kolor znaków (Color, ColorIndex, ColorBi),
* kolor podkreślenia (UnderlineColor),
* kolor znaków diakrytycznych (DiacriticColor),
* odstępy międzyznakowe (Spacing),
* indeksowanie górne (Superscript) i dolne (Subscript), które jednocześnie zmniejsza rozmiar czcionki,
* pozycję (Position) – określa pozycję znaków w pionie względem linii bazowej bez zmiany rozmiaru czcionki.
* podsuwanie znaków (Kerning),
* połączenia znaków (Ligatures),
* stosowanie wielkich liter (AllCaps, SmallCaps),

Podkreślenie może być nie tylko pojedyncze i podwójne, ale też może być linią falistą i wieloma innymi wzorami linii. Odpowiadają za to różne wartości właściwości Underline. Inną opcją jest narysowanie prostokątnego obramowania tekstu z wieloma różnymi stylami linii (Border, Line).

Pewne ustawienia mogą być stosowane tylko do czcionek OpenType. Są to:

* formy numeryczne (NumberForm) i odstępy numeryczne (NumberSpacing),
* alternatywne formy znaków zależne od kontekstu (ContextualAlternates),
* różne formy stylistyczne (StylisticSet).

Obiekt klasy Font ma też właściwości tworzące dodatkowe efekty graficzne, takie jak:

* skalowanie (Scaling),
* uwypuklenie (Emboss) i wytłoczenie (Engrave),
* rysowanie obwiedni (Outline),
* wypełnienie gradientem lub innym wzorem (Fill),
* rozświetlenie (Glow),
* cieniowanie (Shading) i dodawanie cienia (Shadow, TextShadow),
* odbicie lustrzane (Reflection),
* inne efekty trójwymiarowe (ThreeD),
* animacje (Animation).

Tekst może być ukryty (Hidden) i prezentowany z ignorowaniem siatki znaków w wierszu (DisableCharacterSpaceGrid).

Z kolei właściwości obiektu ParagraphFormat określają m.in:

* wyrównywanie tekstu w poziomie (Alignment),
* wyrównywanie linii bazowej tekstu (BaseLineAlignment),
* wcięcie z lewej (LeftIndent, CharacterUnitLeftIndent),
* wcięcie pierwszej linii (FirstLineIndent, CharacterUnitFirstLineIndent),
* wcięcie z prawej (RightIndent, CharacterUnitRightIndent),
* odstęp międzyliniowy (LineSpacing, LineSpacingRule),
* odstęp przed akapitem (SpaceBefore, LineUnitBefore, SpaceBeforeAuto),
* odstęp po akapicie (SpaceAfter, LineUnitAfter, SpaceAfterAuto),
* utrzymywanie akapitu razem z następnym na jednej stronie (KeepWithNext),
* utrzymywanie wierszy akapitu razem na jednej stronie (KeepTogether),
* utrzymywanie skrajnych wierszy akapitu razem na jednej stronie (WidowControl),
* wprowadzanie podziału strony przed akapitem (PageBreakBefore),
* pozycjonowanie tabulacji (TabsStops),
* poziom akapitu w widoku konspektu (OutlineLevel),
* obramowanie akapitu (Borders),
* cieniowanie akapitu (Shading),
* sterowanie podziałami wyrazów (Hyphenation).

Inne właściwości akapitu to:

* automatyczne dopasowanie prawego wcięcia akapitu, gdy jest określona liczba znaków w wierszu (AutoAdjustRightIndent),
* wyrównywanie znaków w akapicie do siatki linii, gdy jest określona liczba linii na stronę (DisableLineHeightGrid),
* wcięcia lustrzane (MirrorIndents) – określa, czy lewe i prawe wcięcie są takie same,
* interpunkcja wysunięta (HangingPunctuation) – umożliwia utrzymywanie znaków interpunkcyjnych takich jak cudzysłowy i nawiasy poza lewą i prawą krawędzią wyrównania akapitu,
* pokazywanie akapitu domyślnie w formie zwiniętej (CollapsedByDefault),
* dodawanie odstępów między znakami pisma dalekowschodniego i literami (AddSpaceBetweenFarEastAndAlpha) lub cyframi (AddSpaceBetweenFarEast­AndDigit),
* połówkowa interpunkcja (HalfWidthPunctuationOnTopOfLine) stosowana dla pisma japońskiego na początku linii,
* sterowanie podziałem linii pisma dalekowschodniego (FarEastLineBreakControl),
* wyłączenie numerowania wierszy (NoLineNumber),
* wybór kierunku czytania z lewej do prawej lub prawej do lewej (ReadingOrder),
* sterowanie zawijaniem tekstu łacińskiego w środku wyrazów (WordWrap, TextBoxTightWrap).

Akapit ma też identyfikator (ID), który jest stosowany, gdy dokument jest zapisywany jako strona Web.

## Tabele

Tabele są reprezentowane przez kolekcję Tables dostępną przez obiekt Document, Range i Selection. Wiersze tabeli są reprezentowane przez kolekcję Rows, kolumny przez kolekcję Columns, a komórki tabeli przez kolekcję Cells. Te kolekcje są dostępne przez obiekty Table, Range, Selection. Kolekcja Tables jest też dostępna dla obiektu Table. Pierwszy element tej kolekcji to dana tabela, a kolejne to tabele zagnieżdżone. Kolekcja Cells jest też dostępna dla obiektów Row i Column.

Właściwości tabeli obejmują:

* identyfikator tabeli (ID) stosowany, gdy dokument jest zapisywany jako strona Web,
* tytuł tabeli (Title),
* opis tabeli (Descr),
* poziom zagnieżdżenia tabeli (NestingLevel),
* pozwolenie na automatyczne dopasowanie rozmiaru komórek do ich zawartości (AllowAutoFit),
* określenie typu autoformatowania (AutoFormatType),
* obramowania tabeli (Borders),
* cieniowanie tabeli (Shading),
* odstępy między zawartością komórek a ich krawędziami (LeftPadding, RightPadding, TopPadding, BottomPadding),
* odstępy między komórkami (Spacing),
* preferowaną szerokość tabeli (PreferredWidth, PreferredWidthType),
* kierunek sortowania komórek w tabeli (TableDirection),
* określenie, czy wszystkie wiersze tabeli mają taką samą liczbę kolumn (Uniform),
* zezwolenie na stosowanie osobnego stylu dla wierszy nagłówkowych (ApplyStyleHeadingRows), dla ostatniego wiersza (ApplyStyleLastRow), dla pasm wierszy (ApplyStyleRowBands), dla pierwszej kolumny (ApplyStyleFirstColumn), dla ostatniej kolumny (ApplyStyleLastColumn), dla pasm kolumn (ApplyStyleColumnBands).

Tytułu (Title) i opisu (Descr) tabeli nie należy mylić z podpisem tabeli (Caption). Podpis jest wstawiany jako osobny akapit oznaczony specjalnym stylem przed lub po tabeli. Tytuł i opis stanowią alternatywne przedstawienie tabeli dla osób niedowidzących. Tytuł może być odczytany najpierw, a później, na żądanie również opis tabeli.

Właściwości wiersza tabeli (Row) określają:

* identyfikator wiersza (ID) stosowany, gdy dokument jest zapisywany jako strona Web,
* obramowania wiersza (Borders),
* cieniowanie (Shading),
* wyrównanie wiersza do lewej/do prawej/do środka (Alignment),
* pozwolenie na podział wiersza między strony (AllowBreakAcrossPage),
* czy wiersz ma być traktowany jako nagłówkowy (HeadingFormat),
* wysokość wiersza (Height, HeightRule),
* poziom zagnieżdżenia (NestingLevel),
* odstępy między kolumnami (SpaceBetweenColumns).

Właściwości kolumny tabeli (Column) zawierają:

* obramowania kolumny (Borders),
* cieniowanie (Shading),
* szerokość kolumny (Width, PreferredWidth, PreferredWidthType),
* poziom zagnieżdżenia (NestingLevel),

Wreszcie właściwości komórki tabeli (Cell) ustalają:

* obramowania komórki (Borders),
* cieniowanie (Shading),
* odstępy między zawartością komórki a jej krawędziami (LeftPadding, RightPadding, TopPadding, BottomPadding),
* czy Word ma wizualnie zmniejszać rozmiar tekstu wprowadzonego do komórki tak, aby tekst pasował do szerokości komórki (FitText),
* wysokość komórki (Height, HeightRule),
* szerokość komórki (Width),
* preferowaną szerokość komórki (PreferredWidth, PreferredWidthType),
* zawijanie tekstu w komórce (WordWrap).
* poziom zagnieżdżenia (NestingLevel).

## Listy

Akapity numerowane i wypunktowane w zakresie mogą być pozyskane przez kolekcję ListParagraphs. Istnieją generalnie dwa typy listy. Pierwszy typ obejmuje automatycznie dodaną liczbę lub punktor na początku każdego akapitu na liście. Drugi typ obejmuje pola LISTNUM, które można umieścić w dowolnym miejscu wewnątrz akapitu. W akapicie może znajdować się więcej niż jedno pole LISTNUM.

Aby dodać formatowanie listy do akapitów, można użyć obiektu ListFormat na obiekcie typu Range poprzez jedną z jego metod:

* ApplyListTemplate(ListTemplate, ContinuePreviousList, ApplyTo, DefaultListBehavior),
* ApplyBulletDefault(DefaultListBehavior),
* ApplyNumberDefault(DefaultListBehavior),
* ApplyOutlineNumberDefault(DefaultListBehavior).

Inne metody obiektu ListFormat to m.in.:

* CanContinuePreviousList(ListTemplate) – sprawdza, czy poprzednia lista może być kontynuowana,
* ConvertNumbersToText (NumberType) – zamienia numerowanie na tekst i dodaje na początku akapitu,
* CountNumberedItems (NumberType, Level) – liczy numerowane lub wypunktowane akapity albo pola LISTNUM w obiekcie ListFormat,
* ListIndent(), ListOutdent() – zwiększa/zmniejsza poziom listy,
* RemoveNumbers(NumberType) – usuwa numerowanie/wypunktowanie z obiektu ListFormat.

Zastosowanie którejś z metod Apply… obiektu ListFormat na akapicie lub sekwencji akapitów powoduje przypisanie ich do obiektu klasy List. Właściwości obiektu ListFormat zwracają informacje o tej liście:

* List – zwraca obiekt List, który reprezentuje pierwszą sformatowaną listę zawartą w określonym obiekcie ListFormat.
* ListLevelNumber – zwraca lub ustawia poziom listy dla pierwszego akapitu w określonym obiekcie ListFormat.
* ListPictureBullet – zwraca obiekt InlineShape reprezentujący obraz używany jako punktor na liście punktorów obrazów.
* ListString – zwraca ciąg znaków reprezentujący wygląd wartości listy pierwszego akapitu w zakresie dla określonego obiektu ListFormat. Na przykład drugi akapit na liście alfabetycznej zwróci B.
* ListTemplate – zwraca obiekt ListTemplate reprezentujący formatowanie listy dla określonego obiektu ListFormat.
* ListType – zwraca typ list zawartych w zakresie dla określonego obiektu ListFormat.
* ListValue – zwraca wartość liczbową pierwszego akapitu w zakresie dla określonego obiektu ListFormat. Na przykład właściwość ListValue zastosowana do drugiego akapitu na liście alfabetycznej zwróci wartość 2.
* SingleList – sprawdza czy określony obiekt ListFormat zawiera tylko jedną listę.
* SingleListTemplate – sprawdza czy cały obiekt ListFormat używa tego samego szablonu listy.

Wszystkie listy utworzone w dokumencie (z wyjątkiem list punktowanych obrazami) mogą być przeglądane poprzez kolekcję Lists tego dokumentu. Pojedyncza lista udostępnia informacje o:

* kolekcji akapitów należących do listy (ListParagraphs),
* szablonie listy pojedynczej (SingleListTemplate),
* nazwie stylu listy (StyleName).

Szablony list (ListTemplates) są dostępne przez galerię list (ListGallery). Istnieją trzy galerie list:

* galeria list wypunktowanych (wdBulletGallery),
* galeria list numerowanych (wdNumberGallery),
* galeria list wielopoziomowych (wdOutlineNumberGallery).

W każdej galerii jest dostępnych siedem predefiniowanych szablonów list (ListTemplates). Z kolei każdy szablon listy (ListTemplate) udostępnia dziewięć poziomów listy (ListLevel). Każdy poziom listy można formatować przez takie właściwości, jak:

* wyrównanie (Alignment),
* czcionka (Font),
* format numerowania (NumberFormat),
* pozycję numerowania (Number Position),
* styl numerowania (NumberStyle),
* znak kończący numerowanie (TrailingCharacter).

Szablon listy wielopoziomowej można skonwertować na szablon listy jednopoziomowej przez metodę Convert.

## Ramki

Ramka (Frame) jest obecnie niezalecanym elementem dokumentu tekstowego, chociaż pozostaje w definicji modelu VSTO. Ramka jest prostokątnym obszarem na stronie, w którym umieszczana jest treść dokumentu niezależna od głównej treści. Ramka ma właściwość Range, która daje dostęp do zawartości ramki. Wszystkie ramki w dokumencie, zakresie lub selekcji udostępnia kolekcja Frames.

Ramka ma następujące właściwości formatujące:

* położenie poziome (HorizontalPosition, RelativeHorizontalPosition),
* położenie pionowe (VerticalPosition, RelativeVerticalPosition),
* wysokość (Height, HeightRule),
* szerokość (Width, WidthRule),
* odległość od tekstu w poziomie (HorizontalDistanceFromText),
* odległość od tekstu w pionie (VerticalDistanceFromText),
* obramowanie ramki (Border),
* cieniowanie (Shading),
* możliwość zablokowania zakotwiczenia (LockAnchor),
* określenie czy tekst jest zawijany dookoła ramki (TextWrap).

## Style

Styl (Style) jest zbiorem właściwości formatujących elementy tekstowe. Dostęp do wszystkich styli w dokumencie jest możliwy poprzez właściwość Styles. Wyróżnia się style wbudowane (ang. built-in styles) i style definiowane przez autora dokumentu. Style wbudowane mogą być redefiniowane w dokumencie lub szablonie dokumentu, ale nie mogą być usuwane. Nazwy styli wbudowanych są ustalone i nie mogą być zmienione przez autora dokumentu, jednak są różne w różnych wersjach językowych aplikacji Word. Dlatego łatwiej się do nich odwoływać przez numery indeksowe kolekcji Styles zdefiniowane przez stałe WdBuiltInStyle. Przykładowe indeksy podano w tab. 1. Style definiowane przez autora dokumentu mają indeksy nieujemne.

Tab. . Przykładowe indeksy styli wbudowanych

|  |  |  |
| --- | --- | --- |
| Stała | Indeks | Znaczenie |
| wdStyleNormal | -1 | Normalny |
| wdStyleHeading1 | -2 | Nagłówek 1 |
| wdStyleHeading2 | -3 | Nagłówek 2 |
| wdStyleNormalIndent | -29 | Wcięcie normalne |
| wdStyleFootnoteText | -30 | Tekst przypisu dolnego |
| wdStyleHeader | -32 | Nagłówek |
| wdStyleFooter | -33 | Stopka |
| wdStyleCaption | -35 | Legenda |
| wdStyleList | -48 | Lista |
| wdStyleListBullet | -49 | Lista punktowana |
| wdStyleListNumber | -50 | Lista numerowana |
| wdStyleTitle | -63 | Tytuł |
| wdStyleDefaultParagraphFont | -66 | domyślna czcionka akapitu |

Istnieją cztery typy styli:

* style znakowe,
* style akapitowe,
* style tabel,
* style list.

Styl znakowy ma określony obiekt Font, który reprezentuje właściwości znakowe, styl akapitowy – obiekt ParagraphFormat, który reprezentuje właściwości akapitowe. Styl akapitowy ma też właściwości:

* NextParagraphStyle – podaje lub ustawia styl następnego akapitu.
* NoSpaceBetweenParagraphsOfSameStyle – określa czy dodawanie odstępu pomiędzy akapitami o tym samym stylu jest wstrzymane.
* Frame – zwraca obiekt definiujący właściwości ramki, w której jest umieszczany akapit.

Styl tablicowy przez właściwość Table udostępnia obiekt typu TableStyle, który reprezentuje właściwości przypisywane do tabel i ich elementów. Styl listy ma właściwość typu ListTemplate określającą szablon listy oraz ListLevelNumber określającą poziom listy.

Style znakowe i style akapitowe mogą być ze sobą powiązane poprzez dwie właściwości klasy Style:

* Linked – określa czy dany styl przechowuje zarówno właściwości akapitowe, jak i znakowe,
* LinkStyle – podaje styl znakowy dla stylu akapitowego.

Kilka właściwości klasy Style jest wspólnych dla różnych typów stylu. Są to właściwości identyfikujące cechy samego stylu:

* NameLocal – nazwa lokalna stylu. Podaje zlokalizowaną nazwę stylu wbudowanego. Dla języka polskiego są to nazwy pokazane w tab. 1.
* Type – określa typ stylu.
* AutomaticallyUpdate – określa czy definicja stylu jest automatycznie aktualizowana po zmianie formatowania wybranego obiektu.
* BaseStyle – styl bazowy, na podstawie którego został zdefiniowany dany styl.
* BuiltIn – określa czy styl jest wbudowany.
* Description – opis stylu (określona przez formatowanie).
* Hidden – określa czy styl jest ukryty na liście styli.
* UnhideWhenUsed – określa czy styl użyty przestaje być ukryty.
* InUse – określa czy styl jest stylem wbudowanym, który został zmodyfikowany lub zastosowany w dokumencie albo na jego podstawie został utworzony nowy styl.
* Locked – czy styl jest zablokowany.
* Priority – podaje liczbę całkowitą określającą kolejność wyświetlania stylu w panelu styli.
* QuickStyle – określa czy styl jest zaliczony do kategorii stylów szybkich.
* Visibility – określa widoczność stylu (do wewnętrznego użycia).

Ponadto jest kilka właściwości określających formatowanie obiektu i zachowanie aplikacji Word.

* LanguageID – identyfikator języka.
* LanguageIDFarEast – identyfikator języka dalekowschodniego.
* NoProofing – określa czy sprawdzanie pisowni i gramatyki jest wstrzymane.
* Borders – kolekcja wszystkich obramowań obiektu.
* Shading – zwraca obiekt definiujący cieniowanie obiektu.

Struktura obiektów Font i ParagraphFormat została omówiona w podrozdziale ‎2.4, obiektu ListTemplate w podrozdziale ‎2.6, a obiektu Frame w podrozdziale ‎2.7. Dodatkowego wyjaśnienia wymaga obiekt klasy TableStyle. Oprócz standardowych właściwości tabeli omówionych w podrozdziale ‎2.5 ma on dwie właściwości: ColumnStripe i RowStripe określające odpowiednio liczbę kolumn w paśmie kolumn i liczbę wierszy w paśmie wierszy, gdy wybrany jest odpowiednio tryb kolumn lub wierszy parzystych/nieparzystych. Ponadto ma metodę Condition(WdConditionCode), która zwraca obiekt formatowania warunkowego fragmentu tabeli (ConditionalStyle). Takie formatowanie warunkowe można zastosować do

* pierwszego wiersza w tabeli,
* ostatniego wiersza w tabeli,
* nieparzystego pasma wierszy w tabeli,
* parzystego pasma wierszy w tabeli,
* pierwszej kolumny w tabeli,
* ostatniej kolumny w tabeli,
* nieparzystego pasma kolumn w tabeli,
* parzystego pasma kolumn w tabeli,
* ostatniej komórki pierwszego wiersza tabeli,
* pierwszej komórki pierwszego wiersza tabeli,
* ostatniej komórki ostatniego wiersza tabeli,
* pierwszej komórki ostatniego wiersza tabeli.

Obiekt ConditionalStyle ma właściwości formatowania ograniczone do:

* obramowań (Borders),
* cieniowania (Shading),
* odstępów między zawartością komórek a ich krawędziami (LeftPadding, RightPadding, TopPadding, BottomPadding),
* właściwości czcionki (Font),
* właściwości akapitowych (ParagraphFormat),

## Rozdziały i podrozdziały

W dokumencie Word nie ma pojęcia rozdziału ani podrozdziału. Są akapity, które mogą mieć różne style, w tym style nagłówkowe (wbudowane w aplikację). Stylem nagłówkowym jest jeden ze styli o numerach indeksowych od -2 do -10 i nazwach (w języku polskim) „Nagłówek 1”… „Nagłówek 9”. Można przyjąć, że ciąg akapitów po akapicie oznaczonym stylem nagłówkowym, stanowi treść rozdziału. Nie ma żadnego sprawdzania poziomów nagłówków, a o tym, czy struktura rozdziałów/podrozdziałów jest hierarchiczna decyduje dyscyplina autorów dokumentu.

## Zakładki

Zarówno obiekt Document, Range, jak i Selection mają właściwość Bookmarks reprezentującą zakładki w tekście. W odróżnieniu od innych elementów dokumentu (akapity, tabele), które tworzą strukturę hierarchiczną, zakładki pojawiają się w dowolnych miejscach i są parowane ze sobą poprzez identyfikatory. Para zakładek tworzy ciągły zakres niezależnie od ich umiejscowienia (np. początek zakładki umieszczony w połowie akapitu przed tabelą i koniec zakładki umieszczony w połowie akapitu za tabelą tworzą zakres obejmujący drugą połowę pierwszego akapitu, tabelę i pierwszą połowę drugiego akapitu – rys. 4).

początek zakładki

koniec zakładki

zakres zakładki

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed tincidunt condimentum diam id ullamcorper. Vivamus volutpat libero ut sem condimentum, at tempus ligula sodales. Donec ornare elit ac ipsum semper luctus.

Curabitur nec tortor nec dolor scelerisque fermentum. Mauris pellentesque, enim sed eleifend dictum, lectus nulla tincidunt sem, vel rutrum libero nulla eu turpis. Proin tincidunt metus sed odio dictum tincidunt. Pellentesque at sem lacus. Nullam eget dui sollicitudin, imperdiet felis at, luctus justo.

Rys. . Ciągły zakres zakładki

## Nagłówki i stopki

Nagłówek jest treścią umieszczaną na górze strony, a stopka – na dole strony. Nagłówki i stopki mogą być definiowane dla każdej sekcji osobno, a w ramach każdej sekcji mogą być różne dla stron parzystych i nieparzystych oraz inne dla pierwszej strony.

Nagłówki i stopki są dostępne przez kolekcje Headers i Footers sekcji. Każda z tych kolekcji zwraca obiekty typu HeaderFooter. Każdy obiekt tego typu ma określony zakres i własne definicje numerowania stron (PageNumbers).

# Rysunki

Aplikacja Word oferuje kilka rodzajów rysunków:

* proste rysunki wektorowe oparte o kolekcję kształtów (Shapes),
* wykresy oparte o dane liczbowe (Charts),
* diagramy różnych typów (Diagrams),
* obrazy (Picture, LinkedPicture),
* grafiki SmartArt.

Elementy graficzne są reprezentowane przez kolekcje Shapes i InlineShapes. Pierwsza kolekcja reprezentuje elementy rozlokowane na powierzchni strony w dowolnych miejscach, przed tekstem, za tekstem lub oblane tekstem, a druga – elementy graficzne przemieszczane razem z tekstem i traktowane tak, jak inne elementy tekstowe.

## Kształty

Obiekt Shape reprezentuje obiekt umieszczany w warstwie rysunkowej. Jest wiele typów obiektu Shape określonych przez wartości MsoShapeType (tab. 2).

Tab. . Typy kształtów

|  |  |  |
| --- | --- | --- |
| Lp. | Stała | Typ |
| 1 | msoAutoShape | autokształt |
| 2 | msoCallout | objaśnienie |
| 3 | msoChart | wykres |
| 4 | msoComment | komentarz |
| 5 | msoFreeform | dowolny kształt |
| 6 | msoGroup | grupa |
| 7 | msoEmbeddedOLEObject | zagnieżdżony obiekt OLE |
| 8 | msoFormControl | formant formularza |
| 9 | msoLine | linia |
| 10 | msoLinkedOLEObject | połączony obiekt OLE |
| 11 | msoLinkedPicture | obraz połączony |
| 12 | msoOLEControlObject | obiekt sterujący OLE |
| 13 | msoPicture | obraz |
| 14 | msoPlaceholder | symbol zastępczy |
| 15 | msoTextEffect | efekt tekstowy |
| 16 | msoMedia | media |
| 17 | msoTextBox | pole tekstowe |
| 18 | msoScriptAnchor | zakotwiczenie skryptu |
| 19 | msoTable | tabela |
| 20 | msoCanvas | płótno |
| 21 | msoDiagram | diagram |
| 22 | msoInk | pismo odręczne |
| 23 | msoInkComment | komentarz odręczny |
| 24 | msoSmartArt | grafika SmartArt |
| 25 | msoSlicer | slicer (stosowany w Excelu) |
| 26 | msoWebVideo | wideo internetowe |

Autokształt jest figurą, która może przyjąć jedną z predefiniowanych form (tab. 3).

Tab. . Niektóre formy autokształtu

|  |  |  |
| --- | --- | --- |
| Lp. | Stała | Forma |
| 1 | msoShapeRectangle | Prostokąt |
| 2 | msoShapeParallelogram | Równoległobok |
| 3 | msoShapeTrapezoid | Trapezoid |
| 4 | msoShapeDiamond | Romb |
| 5 | msoShapeRoundedRectangle | Zaokrąglony prostokąt |
| 6 | msoShapeOctagon | Ośmiokąt |
| 7 | msoShapeIsoscelesTriangle | Trójkąt równoramienny |
| 8 | msoShapeRightTriangle | Right triangle |
| 9 | msoShapeOval | Owal |
| 10 | msoShapeHexagon | Sześciokąt |
| 11 | msoShapeCross | Krzyż |
| 12 | msoShapeRegularPentagon | Pięciokąt |
| 13 | msoShapeCan | Puszka |
| 14 | msoShapeCube | Sześcian |
| 15 | msoShapeBevel | Skos |
| 16 | msoShapeFoldedCorner | Zagięty narożnik |
| 17 | msoShapeSmileyFace | Uśmiechnięta buźka |
| 18 | msoShapeDonut | Pączek |
| 19 | msoShapeNoSymbol | Symbol "Nie" |
| 20 | msoShapeBlockArc | Łuk blokowy |
| 21 | msoShapeHeart | Serce |
| 22 | msoShapeLightningBolt | Błyskawica |
| 23 | msoShapeSun | Słońce |
| 24 | msoShapeMoon | Księżyc |
| 25 | msoShapeArc | Łuk |
| 26 | msoShapeDoubleBracket | Podwójny nawias kwadratowy |
| 27 | msoShapeDoubleBrace | Podwójna klamra |
| 28 | msoShapePlaque | Plakietka |
| 29 | msoShapeLeftBracket | Lewy nawias kwadratowy |
| 30 | msoShapeRightBracket | Prawy nawias kwadratowy |
| 31 | msoShapeLeftBrace | Lewa klamra |
| 32 | msoShapeRightBrace | Prawa klamra |
| 33 | msoShapeRightArrow | Strzałka blokowa skierowana w prawo |

Linia (kształt typu msoLine) może być pojedyncza, podwójna lub kombinowana z cienkiej i grubej.

Objaśnienie (msoCallout) to kształt w formie prostokąta zawierający pole tekstowe i wskazujący inny element graficzny przez linię prostą lub łamaną.

Pole tekstowe (msoTextBox) to kształt w formie tekstowej zawierający napis. Może być stosowane osobno lub umieszczane w autokształtach.

Kształty mogą być grupowane (wówczas tworzą kształt typu msoGroup). Mogą być też umieszczone na płótnie (w kształcie typu msoCanvas), które jest prostokątnym obszarem obejmującym cały rysunek.

Kształty innych typów są stosowane do reprezentowania wykresów, diagramów, obrazów, elementów grafiki SmartArt, a także obiektów OLE i tabel umożliwiając rozmieszczanie ich w dowolnych miejscach na stronie. W ten sposób kształty zastępują elementy typu ramki (Frame), które są wskazywane jako konstrukcja przestarzała.

Kształt jest zawsze dołączony do zakresu zakotwiczenia (ang. anchoring range). Kształt można umieścić w dowolnym miejscu na stronie, która zawiera zakotwiczenie (ang. anchor).

Podobnie jak do dostępu do elementów tekstowych wykorzystuje się elementy typu Range, tak i kształty mogą być dostępne przez element typu ShapeRange, który jest indeksowaną kolekcją kształtów. Do każdego kształtu można się dostać przez jego numer lub nazwę. Gdy kształt jest tworzony, jest jemu przypisywana domyślna nazwa (np. "Rectangle 1"). Oczywiście autor rysunku może przypisać do kształtu bardziej znaczącą nazwę przez właściwość Name kształtu.

Kształt może być też identyfikowany przez identyfikator liczbowy ID i może mieć alternatywny tekst reprezentowany przez właściwość Title. Z kolei właściwość AlternativeText jest wykorzystywana tylko na stronach internetowych.

Kształty są dodawane do kolekcji Shapes dokumentu przez takie metody, jak: AddShape, AddTextbox, AddCallout, AddCurve, AddLabel, AddLine, AddOleControl, AddOleObject, AddPolyline, AddTextEffect, czy też BuildFreeForm. Metoda Group lub Regroup grupuje kształty i zwraca pojedynczy kształt będący nową grupą. Właściwość GroupItems reprezentuje kolekcję elementów grupy, a ParentGroup – wspólną grupę nadrzędną dla kształtu lub zakresu kształtów.

Każdy kształt jest zakotwiczony w pewnym zakresie tekstu. Kotwica jest umieszczana na początku pierwszego akapitu, który zawiera zakres zakotwiczenia. Kształt zawsze pozostanie na tej samej stronie, co jego zakotwiczenie.

Autor może zobaczyć kotwicę ustawiając właściwość ShowObjectAnchors na true. Właściwości Top i Left kształtuokreślają jego położenie pionowe i poziome, Właściwości RelativeHorizontalPositioniRelativeVerticalPosition określają od którego miejsca jest liczone to położenie: czy od początku akapitu zakotwiczenia, czy od kolumny zawierającej akapit zakotwiczenia, czy od marginesu, czy krawędzi strony.

Jeśli właściwość LockAnchor kształtu zostanie ustawiona na true, to kształt nie będzie mógł być przemieszczany na stronie.

## Formatowanie kształtów

Właściwości Width i Height określają bezwzględną szerokość i wysokość kształtu. Można też określić szerokość i wysokość w sposób względny poprzez podanie procentowej wartości dla właściwości WidthRelative lub HeightRelative. Jeśli jest ona ustawiona na -999999 (wdShapeSizeRelativeNone), to liczy się tylko szerokość lub wysokość bezwzględna. Jeśli na inną wartość, to poprzez właściwości RelativeHorizontalSize i RelativeVerticalSize można określić względem czego oblicza się szerokość lub wysokość kształtu: czy rozmiaru marginesu, czy strony.

Jeśli właściwość LockAspectRatio kształtu zostanie ustawiona na true, to proporcje szerokości i wysokości zostaną zachowane przy zmianie jednej z nich.

Właściwość ZOrderPosition określa kolejność prezentowania kształtów (porządek Z), przy czym wartość 1 oznacza kształt na samym spodzie rysunku, a wartość Shapes.Count – na samym wierzchu.

Właściwość Visible ustala czy kształt jest widoczny.

Właściwości formatujące kształt to m.in:

* Line – zwraca obiekt LineFormat reprezentujący właściwości linii,
* Fill – zwraca obiekt FillFormat reprezentujący wypełnienie kształtu zamkniętego,
* Shadow – zwraca obiekt typu ShadowFormat reprezentujący właściwości cienia,
* Glow – zwraca obiekt typu GlowFormat reprezentujący właściwości rozświetlenia,
* Reflection – zwraca obiekt typu ReflectionFormat reprezentujący właściwości odbicia światła,
* SoftEdge – zwraca obiekt typu SoftEdgeFormat reprezentujący właściwości zmiękczenia krawędzi kształtu,
* ThreeD – zwraca obiekt typu ThreeDFormat reprezentujący właściwości efektów trójwymiarowych, takich jak fazowanie krawędzi (Bevel), symulacja grubości (Extrusion), perspektywa, obrót w trzech osiach.
* ModelD – zwraca obiekt typu ModelDFormat reprezentujący właściwości widoku w 3D (np. położenie kamery),
* Callout – zwraca obiekt typu CalloutFormat reprezentujący właściwości kształtów typu msoCallout,
* TextEffect – zwraca obiekt typu TextEffectFormat reprezentujący właściwości efektów tekstowych dla kształtów typu msoWordArt,
* WrapFormat – zwraca obiekt typu WrapFormat reprezentujący właściwości oblewania kształtu tekstem (np. odległość tekstu od kształtu),
* PictureFormat – zwraca obiekt typu PictureFormat reprezentujący właściwości obrazu dla kształtów typu msoPicture i msoLinkedPicture,
* PictureFormat – zwraca obiekt typu PictureFormat reprezentujący właściwości obrazu dla kształtów typu msoPicture i msoLinkedPicture,
* OLEFormat – zwraca obiekt typu OLEFormat reprezentujący właściwości obiektu dla kształtów typu msoEmbeddedOLEObject, msoOLEControlObject i msoLinkedOLEObject,
* LinkFormat – zwraca obiekt typu LinkFormat reprezentujący właściwości powiązania pliku dla kształtów typu msoLinkedOLEObject i msoLinkedPicture.

Właściwości HorizontalFlip i VerticalFlip określają, czy kształt jest prezentowany w odbiciu lustrzanym w poziomie lub w pionie. Właściwość Rotation określa kąt obrotu kształtu.

Właściwość ShapeStyle umożliwia wybór jednego z predefiniowanych styli kształtu, a BackgroundStyle – jednego z predefiniowanych styli tła. Właściwość GraphicStyle wybiera jeden z predefiniowanych styli grafiki SVG.

Właściwość Adjustments reprezentuje kolekcję parametrów liczbowych autokształtu.

Właściwość TextFrame zwraca obiekt typu TextFrame, który reprezentuje właściwości ramki tekstowej kształtu, czyli zakres tekstowy (TextRange) i marginesy tekstu. Z kolei TextFrame2 zwraca kształt typu msoTextBox.

Właściwość HasChart określa, czy kształt zawiera wykres, a wówczas właściwość Chart udostępnia obiekt wykresu.

Właściwość HasSmartArt określa, czy kształt zawiera grafikę SmartArt, a wówczas właściwość SmartArt umożliwia formatowanie tej grafiki.

Właściwość Hyperlink umożliwia przypisanie hiperłącza do kształtu.

Właściwość LayoutInCell określa, czy kształt jest rysowany wewnątrz, czy na zewnątrz tabeli.

Właściwości Nodes i Vertices określają położenia węzłów i krawędzi kształtów typu łamanej, krzywej, swobodnego kształtu.

Właściwość Decorative określa, czy kształt jest oznaczony jako dekoracyjny.

Właściwość CanvasItems daje dostęp do elementów kształtu typu msoCanvas.

Właściwość Child określa, czy dany kształt jest kształtem potomnym (ma właściwość Parent) lub czy wszystkie kształty w zakresie kształtów mają tę samą właściwość Parent.

Właściwość Script reprezentuje blok skryptu lub kodu dla obrazu na stronie internetowej.

## Kształty Inline

Kolekcja InlineShapes reprezentuje te kształty, które nie mogą być swobodnie przesuwane po powierzchni strony, lecz są umieszczone w tekście, w dowolnym miejscu akapitu, pomiędzy znakami. Kształt klasy InlineShape ma takie same właściwości jak kształt klasy Shape z wyjątkiem właściwości określających położenie (Left, Top, RelativeHorizontalPositioniRelativeVerticalPosition), rozmiar względny (WidthRelative, HeightRelative, RelativeHorizontalSize i RelativeVerticalSize) i położenie w wymiarze Z (ZOrderPosition). Ma natomiast właściwości skali (ScaleWidth, ScaleHeight). Oprócz tego ma dodatkowe właściwości:

* Borders – podaje kolekcję obramowania kształtu,
* IsPictureBullet – określa, czy kształt reprezentuje obrazek punktora,
* HorizontalLineFormat – zwraca obiekt formatujący linię poziomą,
* Field – zwraca obiekt klasy Field reprezentujący pole skojarzone z kształtem.

## Wykresy

Obiekt typu Chart reprezentuje wykres oparty o pewien zbiór danych (ChartData). Dane mogą być np. podłączone do arkusza Excel, a mogą być też redagowane w skoroszycie Excela (Workbook) osadzonym w dokumencie Worda. Właściwości wykresów są następujące:

* Walls – zwraca obiekt reprezentujący ściany wykresu 3D.
* BackWall – zwraca obiekt reprezentujący tylną ściany wykresu 3D.
* SideWall – zwraca obiekt reprezentujący boczną ścianę wykresu 3D.
* Floor – zwraca obiekt reprezentujący podłogę wykresu.
* ChartArea – zwraca obiekt reprezentujący obszar wykresu (włącznie z obszarem PlotArea).
* PlotArea – zwraca obiekt reprezentujący obszar rysowania wykresu.
* AutoScaling – określa czy wykres 3D jest skalowany tak, aby zapewnić zbliżony rozmiar do równoważnego wykresu 2D. Właściwość RightAngleAxis musi być ustawiona na true.
* CategoryLabelLevel – zwraca lub ustawia stałą XlCategoryLabel, która określa źródłowy poziom dla etykiet kategorii wykresu.
* ChartColor – zwraca lub ustawia liczbę określającą schemat kolorów wykresu.
* ChartData – zwraca informację o połączonych lub zagnieżdżonych danych wykresu.
* ChartGroups – zwraca obiekt, który reprezentuje pojedynczą grupę wykresów lub kolekcję wszystkich grup wykresów na wykresie.
* ChartStyle – zwraca lub ustawia styl wykresu (typu Variant).
* ChartTitle – zwraca obiekt reprezentujący tytuł wykresu.
* ChartType – zwraca lub ustawia typ wykresu (jako typ XlChartType).
* DataTable – zwraca obiekt reprezentujący widoczną tabelę danych wykresu.
* HeightPercent – zwraca lub ustawia wysokość wykresu 3D jako procent szerokości wykresu (od 5 do 500 procent).
* DepthPercent – zwraca lub ustawia głębokość wykresu 3D jako procent szerokości wykresu (od 20 do 2000 procent).
* DisplayBlanksAs – zwraca lub ustawia sposób w jaki puste komórki są rysowane na wykresie (jedna z wartości XlDisplayBlanksAs).
* Elevation – zwraca lub ustawia pochylenie widoku wykresu 3D (w stopniach).
* GapDepth – zwraca lub ustawia odległość pomiędzy seriami danych na wykresie 3D jako procent szerokości znacznika.
* HasAxis – określa, które osie istnieją na wykresie (typ Variant**)**.
* HasDataTable – określa, czy wykres ma tabelę danych.
* HasLegend – określa, czy wykres ma legendę.
* HasTitle – określa, czy wykres ma widoczny tytuł.
* Legend – zwraca obiekt reprezentujący legendę wykresu
* Perspective – zwraca lub ustawia perspektywę wykresu 3D. Ignorowane, jeśli właściwość RightAngleAxes jest ustawiona na true.
* PivotLayout – nie wspierane przez ten obiekt.
* PlotBy – zwraca lub ustawia wartość xlColumns lub xlRows, która określa czy kolumny, czy wiersze są używane jako serie danych wykresu.
* PlotVisibleOnly – określa, czy tylko widoczne komórki są rysowane.
* RightAngleAxes – określa, czy osie wykresu mają prawe kąty niezależne od obrotu lub pochylenia wykresu.
* Rotation – określa obrót wykresu 3D wokół osi Z (w stopniach).
* SeriesNameLevel – zwraca lub ustawia stałą XlSeriesNameLevel, która określa poziom źródła nazw serii.
* Shapes – zwraca kolekcję kształtów na wykresie.
* ShowAllFieldButtons – określa, czy przyciski wszystkich pól mają być widoczne.
* ShowAxisFieldButtons – określa, czy przyciski pól osi mają być widoczne.
* ShowDataLabelsOverMaximum – określa, czy pokazywać etykiety danych, jeśli wartości są większe od maksimum.
* ShowLegendFieldButtons – określa, czy przyciski pól legendy mają być widoczne.
* ShowReportFilterFieldButtons – określa, czy przyciski pól filtrowania raportu mają być widoczne.
* ShowValueFieldButtons – określa, czy przyciski pól wartości mają być widoczne.

## Diagramy

Aplikacja Word umożliwia tworzenie diagramów kilku różnych typów. Wyróżnia się następujące typy diagramów:

* msoDiagramOrgChart – diagram organizacyjny (w formie drzewa),
* msoDiagramCycle – diagram cykliczny,
* msoDiagramRadial – diagram kołowy,
* msoDiagramPyramid – diagram piramidowy,
* msoDiagramVenn – diagram Venna,
* msoDiagramTarget – diagram docelowy.

Diagramy tych typów są już uważane za przestarzałe i zostały zastąpione przez grafikę typu SmartArt.

## Grafika typu SmartArt

Grafika SmartArt jest nowszym narzędziem do tworzenia diagramów różnego typu. Główne dwie klasy tej grafiki to SmartArt i SmartArtNode. Określenie „Smart” oznacza, że hierarchia węzłów klasy SmartArtNode może być prezentowana w diagramach na wiele różnych sposobów.

Niektóre kategorie diagramów SmartArt to:

* lista,
* proces
* cykl
* hierarchia
* relacje
* macierz
* piramida
* obraz

Ponadto można ściągać inne rodzaje diagramów z Office.com.

W każdej kategorii autor ma do dyspozycji kilkanaście układów diagramów.

Obiekt klasy SmartArt ma następujące właściwości:

* AllNodes - zwraca obiekt typu SmartArtNodes zawierający wszystkie węzły na diagramie grafiki SmartArt.
* Color – zwraca lub ustawia styl kolorów grafiki inteligentnej zastosowany do grafiki SmartArt.
* Layout – zwraca lub ustawia układ grafiki skojarzony z grafiką grafiki Smart Art.
* Nodes – zwraca elementy podrzędne węzła głównego diagramu grafiki SmartArt.
* QuickStyle – zwraca lub ustawia szybki styl grafiki zastosowany do grafiki SmartArt.
* Reverse – zwraca lub ustawia stan diagramu grafiki SmartArt w odniesieniu do (od lewej do prawej) LTR lub (od prawej do lewej) RTL, jeśli diagram obsługuje odwrócenie.

Kolekcja SmartArtNodes zwraca węzły klasy SmartArtNode. Obiekty tej klasy mają poniższe właściwości:

* Hidden – określa, czy węzeł jest ukryty w modelu danych.
* Level – zwraca poziom węzła w hierarchii.
* Nodes – zwraca węzły podrzędne skojarzone z danym węzłem.
* OrgChartLayout – zwraca lub ustawia wartość MsoOrgChartLayoutType skojarzoną z danym węzłem.
* ParentNode – zwraca węzeł nadrzędny skojarzony z danym węzłem.
* Shapes – zwraca zakres kształtów skojarzony z danym węzłem.
* TextFrame2 – zwraca ramkę tekstową skojarzoną z danym węzłem.
* Type – zwraca typ danego węzła (msoSmartartNodeTypeDefault lub msoSmartArtNodeTypeAssistant).

# Wyrażenia matematyczne

Wyrażenia matematyczne są reprezentowane przez obiekty klasy OMath należące do kolekcji OMaths zakresu tekstowego. Wyrażenia mogą być prezentowane w sposób liniowy (jak zwykły tekst) lub strukturalny (jak np. ułamki piętrowe).

Właściwości obiektu OMath są następujące:

* AlignPoint – zwraca lub ustawia liczbę reprezentującą pozycję znakową punktu wyrównania.
* ArgIndex – zwraca lub ustawia liczbę reprezentującą indeks argumentu tego komponentu w odniesieniu do zawierającego obiektu matematycznego.
* ArgSize – zwraca lub ustawia liczbę, która reprezentuje rozmiar pisma argumentu.
* Breaks – zwraca kolekcję OMathBreaks reprezentującą podziały wierszy w wyrażeniu.
* Functions – zwraca kolekcję OMathFunctions reprezentującą funkcje zawarte w wyrażeniu.
* Justification – zwraca lub ustawia wartość sposób wyrównania wyrażenia.
* NestingLevel – zwraca poziom zagłębienia obiektu w wyrażeniu.
* ParentArg – zwraca obiekt OMath, który reprezentuje obiekt nadrzędny lub zawierający argument.
* ParentCol – zwraca obiekt OMathMatCol reprezentujący kolumnę nadrzędną w macierzy.
* ParentFunction – zwraca obiekt OMathFunction reprezentujący nadrzędną funkcję.
* ParentOMath – zwraca obiekt OMath reprezentujący obiekt nadrzędny.
* ParentRow – zwraca obiekt OMathMatRow reprezentujący wiersz nadrzędny w macierzy.
* Range – zwraca zakres tekstowy zawarty w obiekcie.
* Type – zwraca lub ustawia wartość WdOMathType, która określa czy wyrażenie jest wyświetlane jako inline, czy w osobnej linii.

Obiekt OMath może zawierać różne funktory matematyczne:

* OMathAcc – reprezentuje wyrażenie matematyczne z akcentem.
* OMathArgs – reprezentuje kolekcję argumentów wyrażenia.
* OMathBar – reprezentuje wyrażenie matematyczne z kreską nad lub pod.
* OMathBorderBox - Reprezentuje niewidoczny blok wokół równania lub części równania, do którego można przypisać właściwości wpływające na układ lub formatowanie matematyczne całego pola. Na przykład pole może służyć jako emulator operatora z punktem wyrównania lub bez niego, służyć jako podobny punkt przerwania, mieć skojarzony rozmiar argumentu (argSz) lub być zgrupowany tak, aby nie zezwalać na podziały wierszy.
* OMathBox – reprezentuje niewidoczny blok wokół równania lub części równania, do którego można zastosować właściwości wpływające na właściwości matematyczne lub formatowania, takie jak podziały wierszy.
* OMathBreak – reprezentuje podział wiersza w wyrażeniu.
* OMathDelim – reprezentuje obiekt ogranicznika, składający się z separatorów otwierających i zamykających (takich jak nawiasy, nawiasy klamrowe, nawiasy kwadratowe lub pionowe paski) oraz co najmniej jednego elementu zawartego wewnątrz ograniczników.
* OMathEqArray – reprezentuje obiekt tablicy równań matematycznych, składający się z jednego lub większej liczby równań, które mogą być wyrównane w pionie jako jednostka względem otaczającego tekstu w wierszu.
* OMathFrac – reprezentuje ułamek składający się z licznika i mianownika oddzielonych kreską ułamkową. Kreska ułamkowa może być pozioma lub ukośna, w zależności od właściwości ułamka.
* OMathFunc – reprezentuje zastosowanie funkcji lub argumentu w równaniu, które składa się z nazwy funkcji, takiej jak sin lub cos, oraz argumentu.
* OMathFunction – reprezentuje funkcję matematyczną lub strukturę obsługiwaną przez program Microsoft Office Word, taką jak ułamki, całki, sumy i pierwiastki.
* OMathGroupChar – reprezentuje obiekt grupowy, składający się ze symbolu narysowanego nad lub pod tekstem, często w celu wizualnego grupowania elementów.
* OMathLimLow – reprezentuje konstrukcję matematyczną dolnej granicy, składającą się z tekstu na linii bazowej i tekstu o zmniejszonym rozmiarze bezpośrednio pod nim.
* OMathLimUp – reprezentuje konstrukcję matematyczną górnej granicy, składającą się z tekstu na linii bazowej i tekstu o zmniejszonym rozmiarze bezpośrednio nad nią.
* OMathMat – reprezentuje macierz.
* OMathMatCol – reprezentuje kolumnę macierzy.
* OMathMatRow – reprezentuje wiersz macierzy.
* OMathNary – reprezentuje matematyczny obiekt n-elementowy, składający się z obiektu n-elementowego, podstawy (lub operandu) oraz opcjonalnych górnych i dolnych limitów.
* OMathPhantom – reprezentuje matematyczny obiekt n-elementowy, składający się z obiektu n-elementowego, podstawy (lub operandu) oraz opcjonalnych górnych i dolnych limitów.
* OMathRad – reprezentuje matematyczny obiekt pierwiastkowy, składający się z pierwiastka, podstawy i opcjonalnego stopnia.
* OMathScrPre – reprezentuje wyrażenie, które zawiera indeks górny lub dolny po lewej stronie podstawy.
* OMathScrSub – reprezentuje wyrażenie zawierające podstawę i indeks dolny.
* OMathScrSup – reprezentuje wyrażenie zawierające podstawę i indeks górny.
* OMathScrSubSup – reprezentuje wyrażenie zawierające podstawę oraz indeks dolny i górny.

Wprowadzanie wyrażeń ułatwiają obiekty:

* OMathAutoCorrect – reprezentuje możliwość autokorekty matematycznej.
* OMathAutoCorrectEntries – reprezentuje kolekcję pozycji autokorekty
* OMathAutoCorrectEntry – reprezentuje pojedynczą pozycję autokorekty.
* OMathRecognizedFunctions – reprezentuje kolekcję rozpoznawanych funkcji.
* OMathRecognizedFunction – reprezentuje rozpoznawaną funkcję.

# Kontrolki zawartości

Kontrolki zawartości (ang. content controls) to złożone funkcjonalnie elementy, które umożliwiają wprowadzanie i prezentowanie tekstu na podobieństwo formantów formularzy, ale działające w warstwie tekstowej dokumentu. Kolekcja kontrolek (Controls) jest właściwością dokumentu.

Istnieje wiele typów kontrolek zawartości:

* CheckBox – kontrolka opcji reprezentuje dwie wartości: wybrana (selected) i wyczyszczona (cleared). Nie ma specjalnego typu CheckBoxContentControl, ale można utworzyć kontrolkę opcji przez dodanie uogólnionego obiektu ContentControl.
* ComboBoxContentControl – kontrolka wyświetlająca listę elementów do wyboru, ale dająca użytkownikowi możliwość wprowadzania własnych elementów.
* DatePickerContentControl – kontrolka umożliwiająca wybór daty z kalendarza.
* DropDownListContentControl – kontrolka wyświetlająca listę elementów do wyboru bez możliwości wprowadzania własnych elementów.
* GroupContentControl – definiuje region dokumentu, w którego użytkownik nie może edytować ani usunąć. Może zawierać dowolne elementy dokumentu, takie jak tekst, tabele, grafikę i inne kontrolki zawartości.
* PictureContentControl – reprezentuje obraz określony w czasie projektowania lub wykonania. Użytkownik może kliknąć w tę kontrolkę, aby włączyć obraz do dokumentu.
* RichTextContentControl – zawiera tekst i inne element, takie jak tabele, obrazki i inne kontrolki zawartości.
* PlaintTextContentControl – zawiera tylko tekst.
* uogólniona kontrolka zawartości – może reprezentować dowolny typ kontrolki zawartości.

## Wspólne zadania kontrolek zawartości

Pewne wspólne zadania mogą być wykonywane w sposób określony w tab. 4.

Tab. . Wspólne zadania i ich realizacja za pomocą kontrolek zawartości

|  |  |
| --- | --- |
| Aby | Trzeba |
| pobrać lub ustawić wyświetlany tekst | użyć właściwości Text. Kontrolki typów PictureContentControl i ContentControl nie zawierają tej właściwości |
| pobrać lub ustawić tekst tymczasowy, który jest wyświetlany w kontrolce, dopóki użytkownik jej nie zmodyfikuje, kontrolka nie zostanie wypełniona danymi ze źródła danych lub zawartość kontrolki nie zostanie usunięta. | użyć właściwości PlaceholderText. Kontrolka typu PictureContentControl nie zawiera tej właściwości |
| pobrać lub stawić tytuł, który jest wyświetlany w obramowaniu kontrolki zawartości, gdy użytkownik ją kliknie. | użyć właściwości Title. |
| automatycznie usunąć kontrolkę z dokumentu po edycji kontrolki przez użytkownika. (Tekst w formancie pozostaje w dokumencie). | użyć właściwości Temporary. |
| uruchomić kod, gdy użytkownik kliknie kontrolkę zawartości lub gdy kursor zostanie przeniesiony do kontrolki zawartości programowo. | obsłużyć zdarzenie Entering kontrolki. |
| uruchomić kod, gdy użytkownik kliknie poza kontrolką zawartości lub gdy kursor zostanie programowo przeniesiony poza kontrolkę zawartości. | obsłużyć zdarzenie Exiting kontrolki. |
| uruchomić kod po dodaniu kontrolki zawartości do dokumentu w wyniku operacji ponawiania lub cofania. | obsłużyć zdarzenie Added kontrolki. |
| uruchomić kod tuż przed usunięciem kontrolki zawartości z dokumentu. | obsłużyć zdarzenie Deleting kontrolki. |

## Ochrona części dokumentu

Chroniąc część dokumentu, można uniemożliwić użytkownikom zmienianie lub usuwanie zawartości tej części dokumentu. Istnieje kilka sposobów ochrony części dokumentu za pomocą kontrolek zawartości.

Jeśli obszar do ochrony znajduje się wewnątrz kontrolki zawartości, można użyć właściwości kontrolki zawartości, aby uniemożliwić użytkownikom edytowanie lub usuwanie kontrolki:

* Właściwość LockContents uniemożliwia użytkownikom edytowanie zawartości.
* Właściwość LockContentControl uniemożliwia użytkownikom usunięcie kontrolki.

Jeśli obszar do ochrony nie znajduje się wewnątrz kontrolki zawartości lub jeśli trzeba chronić obszar zawierający kontrolki zawartości i inne typy zawartości, można umieścić cały obszar w kontrolce GroupContentControl. W przeciwieństwie do innych kontrolek zawartości, kontrolka tego typu nie udostępnia interfejsu użytkownika, który jest widoczny dla użytkownika. Jej jedynym celem jest zdefiniowanie regionu, którego użytkownicy nie mogą edytować.

## Powiązanie danych z kontrolkami

Dane można wyświetlać w dokumentach, wiążąc kontrolkę zawartości ze źródłem danych. Po zaktualizowaniu źródła danych kontrolka zawartości odzwierciedla zmiany. Możesz również zapisać zmiany z powrotem w źródle danych.

Kontrolki zawartości udostępniają następujące opcje powiązania danych:

* Kontrolki zawartości można powiązać z polami bazy danych lub obiektami zarządzanymi przy użyciu tego samego modelu powiązania danych co Windows Forms.
* Kontrolki zawartości można powiązać z elementami w elementach XML (nazywanych również custom XML parts), które są osadzone w dokumencie.

# Pola

Pole (Field) to fragment dokumentu, który uzyskuje tekst za pomocą pewnego kodu. Główne właściwości pola są następujące:

* Code – zwraca zakres tekstowy reprezentujący kod (formułę) pola.
* Data – zwraca lub ustawia dane w polu ADDIN.
* Index – zwraca liczbę reprezentującą pozycję elementu w kolekcji.
* InlineShape – zwraca obiekt klasy InlineShape reprezentujący obraz, obiekt OLE lub kontrolkę ActiveX będącą wynikiem pola INCLUDEPICTURE lub pola EMBED.
* Kind – zwraca typ linku obiektu Field.
* LinkFormat – zwraca obiekt reprezentujący opcje linku określonego pola, kształtu inline lub kształtu połączonego z plikiem.
* Locked – określa, czy pole jest zablokowane.
* OLEFormat – zwraca właściwości OLE określonego pola.
* Result – zwraca zakres tekstowy reprezentujący wynik pola.
* ShowCodes – określa, czy kody pól są wyświetlane zamiast wyników.
* Type – określa typ pola.

## Informacje o dokumencie

Szereg pól służy do wstawiania do treści dokumentu informacji o tym dokumencie pobieranych z właściwości dokumentu lub innych źródeł. Są to:

* Author – wstawia informację o autorze z właściwości Creator dokumentu.
* Comments – wstawia komentarze z właściwości Comments dokumentu.
* CreateDate – wstawia datę utworzenia z właściwości Created dokumentu.
* Date – wstawia aktualną datę.
* DocProperty – wstawia określoną właściwość dokumentu.
* DocVariable – wstawia określoną zmienną dokumentu.
* FileName – wstawia nazwę pliku dokumentu.
* FileSize – wstawia rozmiar pliku dokumentu.
* Info – wstawia informację o aktywnym dokumencie lub szablonie pobraną z właściwości dokumentu.
* Keywords – wstawia informację o słowach kluczowych z właściwości Keywords dokumentu.
* LastSavedBy – wstawia autora ostatniej modyfikacji dokumentu.
* NumChars – wstawia liczbę znaków dokumentu.
* NumPages – wstawia liczbę stron dokumentu.
* NumWords – wstawia liczbę słów dokumentu.
* Page – wstawia numer strony, na której znajduje się to pole.
* PrintDate – wstawia datę ostatniego drukowania z właściwości Printed dokumentu.
* RevNum – wstawia numer rewizji z właściwości Revision dokumentu.
* SaveDate – wstawia datę i godzinę ostatniego zapisania dokumentu.
* Section – wstawia numer bieżącej sekcji.
* SectionPages – wstawia całkowitą liczbę stron w sekcji. Korzystając z tego pola, należy ponownie rozpocząć numerację stron od 1 w każdej sekcji po pierwszej sekcji.
* Subject – wstawia temat z właściwości Subject dokumentu.
* Template – wstawia nazwę pliku szablonu z właściwości Template dokumentu.
* Time – wstawia aktualny czas do dokumentu.
* Title – wstawia tytuł z właściwości Title dokumentu.
* UserAddress – wstawia adres użytkownika z pola MailingAddress w oknie dialogowym Opcje programu Word.
* UserInitials – wstawia inicjały użytkownika z pola Initials w oknie dialogowym Opcje programu Word.
* UserName – wstawia nazwę użytkownika z pola Name użytkownika w oknie dialogowym Opcje programu Word.

## Numerowanie rysunków i tabel

Do numerowania rysunków, tabel, automatycznego numerowania akapitów etc. służą pola:

* AutoNum – automatycznie numeruje akapity.
* AutoNumLgl – automatycznie numeruje akapity dla publikacji prawnych lub technicznych.
* AutoNumOut – automatycznie numeruje akapity w stylu konspektu.
* ListNum - wstawia numery listy w dowolne miejsca akapitu.
* Seq (Sequence) – sekwencyjnie numeruje rozdziały, tabele, rysunki i inne elementy w dokumencie. Jeśli dodasz, usuniesz lub przeniesiesz element i odpowiadające mu pole Seq, możesz zaktualizować pozostałe pola Seq w dokumencie, aby odzwierciedlić nową sekwencję.

## Indeksy i spisy

Aplikacja Word umożliwia automatyczne tworzenie spisu treści, ułatwia tworzenie indeksów. Służą do tego pola:

* Index – tworzy indeks w oparciu o pola XE.
* XE (Index Entry) – definiuje tekst i numer strony dla pozycji indeksu. Pole XE służy do definiowania elementu, który ma zostać uwzględniony w indeksie.
* RD (Referenced Document) – identyfikuje plik, który ma zostać dołączony podczas tworzenia spisu treści, spisu źródeł lub indeksu z polem TOC, TOA lub Index. Przed zaktualizowaniem pola spisu treści, spisu treści lub indeksu należy ręcznie ustawić początkowe numery stron i wartości sekwencyjne w plikach nazwanych w polach RD. Pole RD nie wyświetla wyniku w dokumencie. Nie można odłączyć pola RD.
* TC (Table of Contents Entry) - definiuje tekst i numery stron dla pozycji w spisie treści oraz na listach spisów, rycin i podobnych treści. Wstaw pole TC bezpośrednio przed tekstem, który chcesz dołączyć do treści.
* TOC (Table of Contents) – tworzy spis treści. Pole spisu treści zbiera wpisy do spisu treści przy użyciu stylów nagłówków, innych określonych stylów, poziomów konspektu, podpisów lub pozycji określonych przez pola TC.
* TA (Table of Authorities Entry) – definiuje tekst i numer strony dla pozycji spisu źródeł.
* TOA (Table of Authorities) – buduje i wstawia spis źródeł. Pole TOA gromadzi wpisy oznaczone polami TA.

## Bibliografia

Bibliografia jest alfabetyczną listą źródeł bibliograficznych. Pole Citation wstawia informację o określonym źródle bibliograficznym w miejscu jego cytowania. Umożliwia automatyczne przenumerowanie cytowań. Do jednego źródła można odwoływać się wielokrotnie. Pole Bibliography tworzy alfabetyczną listę źródeł bibliograficznych i wyświetla ją w miejscu wstawienia tego pola.

## Hiperłącza i referencje

Pole Hyperlink wstawia do dokumentu hiperłącze, które umożliwia przeniesienie się użytkownika do innego miejsca w tym samym dokumencie, w innym dokumencie lub otwarcie wskazanego pliku albo URL.

Jest też pole GoToButton, które umożliwia tylko przeniesienie się użytkownika do innego miejsca w tym samym dokumencie, dlatego bardziej wskazane jest używanie pola Hyperlink.

Pozostałe pola, wymienione niżej wstawiają do dokumentu treści, to których się odwołują, ale dzięki opcji /h mogą się też zachowywać jak hiperłącza:

* NoteRef – wstawia znacznik odwołania do przypisu dolnego lub końcowego, który został oznaczony zakładką, aby utworzyć wiele odwołań do tej samej notatki lub do odsyłaczy do przypisów dolnych lub końcowych. Jeśli zmieni się kolejność przypisów dolnych lub końcowych, nowy to wynik pola NoteRef będzie odzwierciedlał nową numerację.
* PageRef – wstawia numer strony zakładki dla odsyłacza.
* Ref – wstawia tekst lub grafikę reprezentowaną przez określoną zakładkę. Zakładka musi być zdefiniowana w aktywnym dokumencie.
* StyleRef – wstawia tekst sformatowany przy użyciu określonego stylu. Po wstawieniu do nagłówka lub stopki pole StyleRef drukuje pierwszy lub ostatni tekst sformatowany przy użyciu określonego stylu w treści dokumentu bieżącej strony, umożliwiając wydrukowanie nagłówków lub stopek w stylu słownikowym.

## Obiekty OLE i bazy danych

Dokument Worda korzysta z technologii OLE przez pola:

* Embed - zagnieżdża w dokumencie obiekt OLE z innej aplikacji.
* Link – łączy dokument z obiektem OLE z innej aplikacji przez plik.

Pole Database wstawia wyniki kwerendy do bazy danych w tabeli programu Word. Zawiera wszystkie informacje potrzebne do nawiązania połączenia z bazą danych i wykonania kwerendy języka SQL. Aktualizacja pola powoduje ponowne wysłania zapytania do bazy danych.

## Korespondencja seryjna

Do korespondencji seryjnej służą następujące pola:

* AddressBlock – wstawia blok adresowy w korespondencji seryjnej.
* GreetingLine – wstawia linię powitania w korespondencji seryjnej.
* MergeField – wyświetla nazwę pola danych w podwójnych cudzysłowach ostrych w dokumencie głównym korespondencji seryjnej, na przykład «Imię». Gdy dokument główny jest scalany z wybranym źródłem danych, w miejsce pola korespondencji seryjnej wstawiane są informacje z określonego pola danych.
* MergeRec – w scalonym dokumencie wyświetla pozycję porządkową bieżącego rekordu danych. Liczba odzwierciedla sortowanie lub filtrowanie zastosowane do źródła danych przed scaleniem. Za pomocą pola MergeRec można ponumerować każdy scalony dokument w korespondencji seryjnej w kolejności, w jakiej posortowano źródło danych.
* MergeSeq – numeruje każdy scalony rekord w korespondencji seryjnej. Ten numer nie jest widoczny, dopóki nie zakończysz scalania. Liczba może różnić się od wartości wstawianej przez pole MergeReq.
* Next – instruuje program Word, aby scalił następny rekord danych z bieżącym scalonym dokumentem, zamiast rozpoczynać nowy scalony dokument. Pole Next służy do drukowania określonej liczby rekordów danych w jednym scalonym dokumencie.
* NextIf – porównuje dwa wyrażenia. Jeśli porównanie jest prawdziwe, program Word scala następny rekord danych z bieżącym dokumentem korespondencji seryjnej. Jeśli porównanie jest fałszywe, program Word scala następny rekord danych z nowym dokumentem korespondencji seryjnej.
* SkipIf – porównuje dwa wyrażenia: jeśli porównanie jest prawdziwe, funkcja SkipIf anuluje bieżący dokument korespondencji seryjnej, przechodzi do następnego rekordu danych w źródle danych i rozpoczyna nowy dokument korespondencji seryjnej; jeśli porównanie jest fałszywe, program Microsoft Word kontynuuje bieżący dokument korespondencji seryjnej.

Pozostałe dwa pola mogą być wykorzystywane zarówno w korespondencji seryjnej, jak i w interakcji z użytkownikiem:

* Ask – powoduje, że Word prosi użytkownika o wprowadzenie informacji podając określoną wskazówkę.
* Fill-In – prosi użytkownika o wprowadzenie tekstu za każdym razem, gdy to pole jest aktualizowane, na przykład po naciśnięciu F9 lub przejściu przez rekordy w korespondencji seryjnej. Podczas tworzenia nowego dokumentu opartego na szablonie (pliku dotm lub dotx) zawierającego pola do wypełnienia program Word monituje o odpowiedź w każdym polu do wypełnienia.

## Inne pola

Poniższe pola wstawiają do dokumentu inne treści:

* IncludePicture – wstawia obrazek z określonego pliku
* IncludeText – wstawia tekst i grafikę z określonego pliku lub dokumentu.
* Quote – wstawia tekst literalny do dokumentu.
* Symbol – wstawia znak o podanym kodzie.

Jest jeszcze kilka innych pól:

* Advance – przesuwa punkt początkowy następującego po polu tekstu w prawo lub w lewo, w górę lub w dół albo do określonej pozycji poziomej lub pionowej.
* MacroButton – wstawia komendę uruchomienia makro.
* Print – wysyła znaki kodu sterującego drukarką do wybranej drukarki. Program Microsoft Word wyświetla wynik tylko wtedy, gdy dokument jest drukowany.
* Private – przechowuje dane dla dokumentów przekonwertowanych z innych formatów plików. Program Microsoft Word wstawia pole Private podczas konwertowania formatów plików; pole to zawiera dane potrzebne do konwersji dokumentu z powrotem do oryginalnego formatu pliku.

# Redagowanie dokumentu

Redagowanie dokumentu oznacza kilka działań, które mogą być podejmowane za pomocą aplikacji Worda:

* sprawdzanie pisowni i gramatyki,
* wprowadzanie komentarzy,
* rejestrowanie zmian w dokumencie,
* współautorstwo dokumentu,
* edytowanie dużych dokumentów.

## Sprawdzanie pisowni i gramatyki

Jeśli język zakresu tekstowego jest określony (przez właściwość LanguageID, LanguageIDFarEast lub LanguageIDOther) i jego właściwość NoProofing nie została ustawiona na true, to aplikacja Worda dokonuje w tle sprawdzenia pisowni i gramatyki tego zakresu w określonym języku. Po zakończeniu sprawdzania ustawiane są dwie właściwości: SpellingChecked i GrammarChecked oznaczające, że zostało zakończone sprawdzanie odpowiednio pisowni i gramatyki. Wówczas aktualne są dwie kolekcje błędów: SpellingErrors i GrammaticalErrors. Można zażądać ponownego sprawdzenia zakresu ustawiając właściwości SpellingChecked lub GrammarChecked na false. Obie kolekcje błędów są typu ProofreadingErrors. Zwracają one obiekty typu Range, które reprezentują błędne fragmenty tekstu.

Można wymusić sprawdzanie pisowni i gramatyki z kodu VSTO wykorzystując metody CheckSpelling i CheckGrammar. Można też wymusić sprawdzanie synonimów metodą CheckSynonyms. W przypadku błędów pisowni można pobrać sugestie poprawnych słów przez metodę GetSpellingSugesions.

Jeśli język nie jest znany, to można uruchomić metodę DetectLanguage, która rozpoznaje język znak po znaku. Po jej zakończeniu ustawiana jest na true właściwość LanguageDetected.

## Komentarze

Komentarze przypominają konstrukcją zakładki. Kolekcja Comments dotyczy dokumentu, zakresu tekstowego, selekcji. Podobnie jak zakładki, komentarze mogą być wstawiane w dowolnym miejscu i mogą obejmować dłuższy fragment tekstu, wiele akapitów, tabele, komórki tabeli. W odróżnieniu od zakładek nie ma właściwości dających dostęp do początku i końca, ale za to mają właściwość Scope zwracającą zakres tekstu objęty komentarzem. Z kolei właściwość Range podaje treść komentarza.

Inne właściwości obiektu Comment to:

* Author – właściwość przestarzała, nie wskazana do stosowania.
* Contact – podaje obiekt klasy CoAuthor reprezentująca autora. który wprowadził dany komentarz.
* Date – podaje datę utworzenia komentarza.
* Done – podaje lub ustawia wartość true, gdy komentarz został zamknięty.
* Initial – właściwość przestarzała, nie wskazana do stosowania.
* IsInk – określa, czy komentarz został napisany odręcznie.
* Reference – zwraca obiekt Range, który reprezentuje przypis dolny, przypis końcowy lub znak referencyjny komentarza.
* Replies – reprezentuje kolekcję komentarzy podawanych jako odpowiedź do tego komentarza.
* ShowTip – właściwość przestarzała, nie wskazana do stosowania.

## Rejestrowanie zmian

Jeśli właściwość TrackRevisions jest ustawiona na true, to zmiany są rejestrowane w ramach danej rewizji. Pojedyncza zmiana jest reprezentowana przez obiekt klasy Revision. Zmiany mogą być następujących typów:

* wdRevisionDelete – usunięcie zakresu,
* wdRevisionInsert – wstawienie zakresu,
* wdRevisionParagraphProperty – zmiana właściwości akapitu,
* wdRevisionReconcile – rozwiązano konflikt,
* wdRevisionSectionProperty – zmiana właściwości sekcji,
* wdRevisionStyleDefinition – zmiana definicje stylu,
* wdRevisionConflict – pojawił się konflikt,
* wdRevisionDisplayField – zmiana wyświetlania pola,
* wdRevisionParagraphNumber – zmiana numeru akapitu,
* wdRevisionProperty – zmiana właściwości,
* wdRevisionReplace – zamiana,
* wdRevisionStyle – zmiana stylu,
* wdRevisionTableProperty – zmiana właściwości tabeli.

Obiekt Revision udostępnia następujące informacje:

* Author – oznaczenie autora, który dokonał zmiany,
* Date – datę i czas zmiany,
* Type – typ zmiany,
* Range – zakres zmiany,
* FormatDescription –opis przy zmianie formatowania,
* Style – zmieniony styl,
* MovedRange – przesunięty zakres tekstowy,
* Cells – kolekcja komórek zaznaczonych markerami zmiany.

## Współautorstwo

Dokument może być jednoczenie redagowany w sieci WWW przez wielu autorów. Rejestruje to obiekt CoAuthoring powiązany z dokumentem. Obiekt ten udostępnia następujące właściwości:

* Authors – podaje kolekcję obiektów klasy CoAuthors informujących o autorach aktualnie edytujących dokument.
* CanMerge – określa, czy dokument może zostać automatycznie połączony z oryginałem.
* CanShare – określa, czy dokument może być współdzielony.
* Conflicts – podaje kolekcję obiektów klasy Conflict reprezentujących konflikty między zmianami.
* Locks – podaje kolekcję obiektów klasy CoAuthorLock reprezentujących blokady w dokumencie.
* Me – podaje obiekt CoAuthor reprezentujący bieżącego użytkownika.
* PendingUpdates – określa, czy dokument ma aktualizacje, które nie mogą być zaakceptowane.
* Updates – podaje kolekcję obiektów typu CoAuthUpdates reprezentujących aktualizacje dostępne dla dokumentu.

Obiekt Conflict reprezentuje konflikty między zmianami dokonanymi przez różnych autorów. Wyróżnia się więcej typów konfliktów niż w samych rewizjach. Określa to tab. 5.

Tab. . Typy konfliktów

|  |  |  |
| --- | --- | --- |
| Lp. | Typ | Znaczenie |
| 1 | wdRevisionInsert | Wstawienie |
| 2 | wdRevisionDelete | Usunięcie |
| 3 | wdRevisionProperty | Zmiana właściwości |
| 4 | wdRevisionParagraphNumber | Zmiana numeru właściwości |
| 5 | wdRevisionDisplayField | Zmiana wyświetlania pola |
| 6 | wdRevisionReconcile | Zmiana oznaczona jako konflikt rozwiązany |
| 7 | wdRevisionConflict | Zmiana oznaczona jako konflikt |
| 8 | wdRevisionStyle | Zmiana stylu |
| 9 | wdRevisionReplace | Zamiana |
| 10 | wdRevisionParagraphProperty | Zmiana właściwości akapitu |
| 11 | wdRevisionTableProperty | Zmiana właściwości tabeli |
| 12 | wdRevisionSectionProperty | Zmiana właściwości sekcji |
| 13 | wdRevisionStyleDefinition | Zmiana definicji stylu |
| 14 | wdRevisionMovedFrom | Przesunięto zawartość stąd |
| 15 | wdRevisionMovedTo | Przesunięto zawartość dotąd |
| 16 | wdRevisionCellInsertion | Wstawiono komórkę tabeli |
| 17 | wdRevisionCellDeletion | Usunięto komórkę tabeli |
| 18 | wdRevisionCellMerge | Złączono komórkę tabeli |
| 19 | wdRevisionCellSplit | Zarezerwowano do wewnętrznego użycia |
| 20 | wdRevisionConflictInsert | Zarezerwowano do wewnętrznego użycia |
| 21 | wdRevisionConflictDelete | Zarezerwowano do wewnętrznego użycia |

## Edytowanie dużych dokumentów

Duże dokumenty mogą być redagowane partiami. Trzeba wówczas edytować osobne dokumenty, utworzyć dokument główny i wstawić te osobne dokumenty do głównego jako dokumenty składowe (ang. subdocuments). Dokument główny ma ustawioną właściwość IsMasterDocument. Dokument składowy ma ustawioną właściwość IsSubdocument. Dokument główny ma kolekcję Subdocuments która reprezentuje dokumenty składowe tego dokumentu. Każdy dokument składowy ma właściwości:

* HasFile – określa, czy dokument składowy został zapisany w pliku.
* Level – określa poziom nagłówka użyty do utworzenia dokumentu składowego.
* Locked – określa, czy dokument składowy jest zablokowany w dokumencie głównym.
* Name – podaje lub ustawia nazwę określonego obiektu.
* Path – podaje ścieżkę pliku na dysku lub zasobu w sieci Web.
* Range – podaje zakres tekstowy reprezentujący fragment dokumentu, który jest zawierany w określonym obiekcie.

# Podsumowanie

W tym dokumencie przedstawiono ogólną strukturę modelu obiektowego dokumentu Worda. Skoncentrowano się na właściwościach obiektów interfejsowych dostępnych w modelu VSTO, a większość metod pominięto, gdyż celem analizy było raczej zorientowanie się, jakie informacje można pozyskać z dokumentu niż przygotowanie do pisania dodatku do aplikacji Worda.

Z doświadczenia i eksperymentów przeprowadzonych przy okazji tworzenia tego opracowania wynikają pewne wnioski, które tu przedstawiono poniżej:

1. Model opisuje interfejs programowy aplikacji (API) Worda, który został utworzony w technologii COM (Common Object Model) dziś uważanej za przestarzałą.
2. Podstawowym elementem tego model jest interfejs; w modelu zdefiniowano kilkaset interfejsów.
3. Główny interfejs Application jest implementowany przez ko-klasę (ang. coclass) ApplicationClass. Instancję tej klasy pozyskuje się z systemu, w którym uruchomiona jest aplikacja Worda. Następnie wywołując metody tej klasy uzyskuje się instancje implementujące pozostałe interfejsy.
4. Obiekt klasy ApplicationClass i pozostałe obiekty są tworzone w warstwie natywnej aplikacji Worda, a do warstwy kodu zarządzanego aplikacji .NET przekazywane są interfejsy tych obiektów. Konsekwencją takiego mechanizmu jest różnica w zarządzaniu pamięcią obiektów między warstwą natywną i zarządzaną. Czas życia obiektów z warstwy natywnej jest uzależniony od aplikacji Worda, a czas życia obiektów z warstwy zarządzanej – od mechanizmu garbage collection aplikacji .NET. Dlatego obiektów pozyskiwanych z aplikacji Worda nie należy traktować tak samo, jak obiektów zarządzanych. Nie można liczyć na to, że obiekty te będą istniały przez dłuższy czas. Wywołanie metod tych obiektów może spowodować wyjątek związany z brakiem dostępu do obiektów natywnych.
5. Wywoływanie metod natywnych z aplikacji .NET zajmuje więcej czasu niż wewnątrz warstwy zarządzanej i niż wewnątrz warstwy natywnej. Różnica wynika przede wszystkim z innej implementacji parametrów typu String. Dlatego programy działające w dużej mierze w pętli pobierającej dane z aplikacji Word, przetwarzającej je w procedurach .NET i odsyłające je z powrotem, nie działają efektywnie wystarczająco do wymagań współczesnych użytkowników.
6. W niektórych zastosowaniach bardziej efektywne jest wykorzystanie interfejsu Find aplikacji Worda, który umożliwia wyszukiwanie i zamianę tekstu, w tym właściwości formatowania.
7. Aplikacje, które mają przeglądać treść dokumentu i stosować bardziej zaawansowane algorytmy do przetwarzania jego fragmentów musiałyby działać w oparciu o kolekcję Characters, która podaje obiekty typu Range dla każdego znaku. Ten mechanizm byłby mocno nieefektywny.
8. W takich zastosowaniach wskazane jest działanie na pliku docx dokumentu. Format takiego pliku jest opisany w standardzie DocumentFormat.OpenXml, który jest oparty na strukturze XML.

1. https://learn.microsoft.com/en-us/previous-versions/dotnet/netframework-4.0/aax7sdch(v=vs.100) [↑](#footnote-ref-1)
2. https://www.ietf.org/rfc/rfc2413.txt [↑](#footnote-ref-2)