

# Jak przeżyć w nieprzyjaznym środowisku WYSIWYG

Wojciech Myszka

## Streszczenie

Gdy już się trochę nauczymy L<sup>A</sup>T<sub>E</sub>Xa przychodzi taki dzień, że dostajemy piękną przygotowaną – w Wordzie – i wydrukowaną – na kartce papieru – instrukcję [3] mówiącą jak artykuł (dzieło) ma wyglądać.

Pojawia się problem skonfrontowania wszystkich wymagań z naszą znajomością systemu: Jaką klasę wybrać? Jak się font Computer Modern ma do zalecanego Times New Roman? I ogólniej – z jakich fontów korzystać? Jak zmodyfikować „standardowe klasy” aby otrzymany rezultat, w możliwie największym stopniu, spełniał wymagania PT Redakcji.

I o tym (na przykładzie klas mwcls [7]) będzie mowa. Z (dokładnie) rozpisanymi przykładami i znaczeniem modyfikowanych parametrów.

A na koniec parę słów do czego cała praca, którą wykonaliśmy się nada. . .

## 1 Wstęp

Celem tego pracy, bardziej niż przedstawienie gotowego rozwiązania, jest zachęcenie czytelnika do własnych prób modyfikowania „standardowych” klas dokumentów w celu dostosowania **wyglądu** do wymagań redakcyjnych.

Zwracam uwagę, że standardowa licencja [4] zazwyczaj na takie modyfikacje zezwala – nakazując jednak aby wersja zmodyfikowana była rozpowszechniana jako osobne, nowe dzieło. Tak licencja, jak i dobry obyczaj nakazują, zaznaczenie wprowadzonych zmian.

W bardzo wielu przypadkach nie ma potrzeby bezpośredniego modyfikowania klasy, z której korzystamy. Możemy utworzyć klasę „pochodną” (w zarysie, jest to opisane po polsku w [7]).

Oparłem się na Instrukcji. . . ([3]), która wpadła w moje ręce. Nie jestem już teraz pewien z jakiego wydawnictwa ona pochodzi. Nie jest to jednak specjalnie istotne. Bardzo wiele wydawnictw (czy nawet konferencji), które dbają o jednolity wygląd publikowanych materiałów takie instrukcje rozpowszechnia. To, co mnie najbardziej dziwi, to fakt, że Autorzy tych instrukcji niezmiernie rzadko dostarczają plik stylu dokumentu edytora MS Word.

Cytaty z instrukcji będą zaznaczane kursywą.

*Referat należy pisać dowolną wersją edytora tekstu Word for Windows. . .* – z tym sobie najprawdopodobniej nie poradzimy. L<sup>A</sup>T<sub>E</sub>X to jednak nie Word. Może i dobrze.

Opracowaną zgodnie z zamieszczonymi radami klasę (alaward.cls) i plik konfiguracyjny T<sub>E</sub>X4ht (alaward.cfg) umieszczam na swojej stronie: <http://www.immt.pwr.wroc.pl/~myszka/TeX/BachoTe>

*Słowa kluczowe pisać czcionką Times New Roman CE, wielkość pisma 9 punktów, odstęp między wierszami przynajmniej 11 punktów, pismo pochyle (kursywa)*

Imię i NAZWISKO (11punktów)

**INSTRUKCJA DLA AUTORÓW ARTYKUŁÓW**  
(tytuł referatu 13 punktów, odstęp między wierszami przynajmniej 15 punktów, pogrubiona)

Instrukcja prezentuje układ merytoryczny oraz techniczne wymagania dotyczące referatów przeznaczonych do druku metodą fotooffsetową. Referat należy pisać dowolną wersją edytora tekstu Word for Windows i przesyłać w formie wydruku z drukarki laserowej na papierze oraz pliku na dyskiecie (FDD 1.44MB). Format kolumny (pole zadruku **bez numeru strony** wynosi 13.5×19 cm. Przed tekstem referatu omówienie (10–15 wierszy) problemu, metod i warunków jego rozwiązania. Omówienie to (abstract) należy pisać na szerokość 13 cm odsunięte od lewego marginesu o 0.5 cm (pierwszy wiersz z wcięciem akapitowym 0.5 cm) krojem pisma Times New Roman CE wielkość pisma 9 punktów, odstęp między wierszami przynajmniej 11 punktów.

X2006. Zawarte tam komentarze zawierają dodatkowe informacje i wskazówki.

## 2 Format strony

W tej części opiszę wszystko to co praktycznie nie wymaga zmian kodu klasy.

To co jest u samej góry strony (tytuł, autorzy, słowa kluczowe, itd.) nie zostało opisane w sposób szczególny. Zostało przedstawione jedynie na ilustracji.

Są to informacje, które w terminologii L<sup>A</sup>T<sub>E</sub>Xowej generowane są za pomocą polecenia `\maketitle` (za wyjątkiem słów kluczowych). Przekonujemy zatem (i rozbudujemy) to polecenie tak aby spełnić wszelkie wymagania.

*Format kolumny (pole zadruku bez numeru strony wynosi 13,5 × 19 cm.* – Użyjemy zatem pakietu geometry z następującymi parametrami:

```
\usepackage[a4paper,width=13.5cm,height=19cm]{geometry}
```

Przed tekstem referatu omówienie (10–15 wierszy) [...]. Omówienie to (abstract) należy pisać na szerokość 13 cm odsunięte od lewego marginesu o 0,5 cm (pierwszy wiersz z wcięciem akapitowym 0,5 cm) krojem pisma Times New Roman CE (wielkość pisma 9 punktów, odstęp między wierszami przynajmniej 11 punktów). Omówienie zatem należy wstawić w specjalne środowisko, nazwiemy je streszczenie (zachowując sobie prawo do użycia nazwy abstract na streszczenie po angielsku).

```
\newenvironment{streszczenie}
  {\list{}{\leftmargin0.5cm%
  \rightmargin0pt}\secondarysize%
  \listparindent0.5cm%
  \itemindent0.5cm\item\relax}
  {\endlist}
```

Strony nie powinny być numerowane – zatem używamy polecenia `\pagestyle{empty}`.

Czcionka Times New Roman – zostanie zastąpiona czcionką QTimes. Ewentualnie można użyć pakietu `winfonts`, który udostępni kilka, udostępnianych wraz z systemem Windows XP, czcionek: Arial, Franklin Gothic, Tahoma, Comic Sans, Georgia, Times, Courier New, Palatino, Linotype, Verdana, Wingding oraz Webdings.

### 3 Czcionka

Tekst główny referatu należy pisać czcionką 11 pt., odstęp między wierszami przynajmniej 13 punktów, [...] wcięcie akapitowe – 0,5 cm...

To jest bardzo proste: użyjemy klasy `mwart` [7], przyjmując jako podstawową wielkość fontu 11pt:

```
\documentclass[11pt]{mwart}
```

gdzie, niestety, font podstawowy zdefiniowany jest (plik `mw11.clo`) w sposób następujący:

```
\@setfontsize\normalsize\@xipt{13.6}
```

w powyższym `\@xipt` zostało zdefiniowane w pliku `latex.ltx` jako

```
\def\@xipt{10.95}
```

być może zatem lepiej będzie przeddefiniować polecenie `\normalsize`:

```
\@setfontsize\normalsize{11}{13}
```

Słowa kluczowe (w tytule), przypisy, tytuły podrozdziałów, podpisy rysunków i tabel, bibliografia – powinny być składane fontem o wielkości 9 punktów oraz odstępem między wierszami 11 punktów. Plik `mw11.clo` definiuje polecenie `\footnotesize` jako `\@setfontsize\footnotesize\@ixpt{11}` (tym razem `\@ixpt` to 9!). Zatem może zostać.

Na koniec – tytuł pracy powinien być składany fontem 13/15 punktów. Niestety, żadna ze standardowych wielkości nie pasuje, musimy więc jedną

przeddefiniować. Najbliższe są `\large` – 12/14 punktów i `\Large` – 14,4/18. Modyfikujemy `\large`:

```
\@setfontsize\large{13}{15}
```

Wzory matematyczne składać pismem 11 p. na osi kolumny z numerem wzoru w okrągłym nawiasie dosuniętym do prawego brzegu kolumny. Symbole zmiennych we wzorach i w tekście należy pisać pismem pochyłym. Zatem standardowo. Przed i po wzorze powinny być odstępy 0,5 cm (zwracam uwagę, że nasi redaktorzy niechętnie dopuszczają skład wzorów, w którym odległość wzoru od „krótkich” linii jest inna niż w przypadku linii pełnych. Definicje odstępów przed i po wzorach wystawionych umieszczone są razem z definicją standardowego (*normalsize*) fontu dokumentu!

Tabele łącznie z tytułem składać pismem 9 p. – klasy `mwcls` przewidują taką sytuację: jeżeli zostaną użyte z nieobowiązkowym parametrem `floatssmall` tabele i podpisy składane są krojem zdefiniowanym jako `\secondarysize` (standardowo `\small`) – zatem też powinniśmy zmodyfikować.

```
\def\secondarysize{\footnotesize}
```

Ponieważ wcięcie akapitowe ma wynosić 5 mm zatem zamieniamy polecenie

```
\setlength\parindent{1.5em}
```

na

```
\setlength\parindent{0.5cm}
```

### 4 Podpisy pod rysunkami

Instrukcja nakazuje stosowanie dwujęzycznych podpisów tabel i rysunków. Sprawia to pewien kłopot, gdyż wymaga dosyć daleko idących ingerencji w użytą klasę (`mwcls`). Istnieje, co prawda, pakiet `ccaption` [5], ale bezpośrednio jego użycie również jest trudne: zakłada on użycie „standardowych” klas dokumentów. Autor pakietu, co prawda, uwzględnił istnienie kilku innych – również niestandardowych – pakietów: `subfigure` czy `caption2`, ale nie wie nic o `mwcls`. . . Decyzja projektowa powinna być poprzedzona głębokim namysłem gdyż korzystając z `ccaption` automatycznie rezygnujemy z części możliwości oferowanych przez `mwcls`.

Na potrzeby tej pracy – modyfikuję polecenie `\caption` dodając mu dodatkowy argument obowiązkowy – podpis angielski. Wadą tego rozwiązania jest to, że **musimy** do każdej ilustracji i tabeli dodawać podpis angielski.

Konstrukcja podpisu (jednakowe polecenie dla środowiska `figure` i `table`) jest dosyć rozbudowana.

Polecenie jest sparametryzowane. Podstawowe parametry definiowane są wraz z odpowiednim otoczeniem. Samo polecenie `\caption` wywołuje najpierw polecenie `@caption`, a ostateczne formatowanie podpisu realizuje polecenie `@makecaption`. Wejdzimy w środek tej konstrukcji, dwukrotnie używając `@makecaption` z różnymi parametrami (szczegóły w pliku klasy).

## 5 Tytułaria

Trzeba przyznać, że klasy `mwcls` bardzo ułatwiają manipulowanie wyglądem tytułariów. Wymagania Wydawcy są następujące: tytuł rozdziału powinien być centrowany, składany wersalikami, czcionką 11/13 punktów. Odstęp przed tytułem ma wynosić 1 cm, a po 0,7 cm.

Wszystkie te wymagania pozwala zrealizować następujące polecenie (dokładnie opisane w dokumentacji [7]):

```
\SetSectionFormatting{section}
  {1cm}
  {\FormatBlockHeading{\centering%
    \normalsize\MakeUppercase}}
  {0.7cm}
```

Ponieważ nie znoszę manieri pisania tytułariów wersalikami polecenie `\MakeUppercase` pozwala pozostać w zgodzie i z własnymi przyzwyczajeniami i z wymaganiami Wydawcy.

Tytuły podrozdziałów (`\subsection`) składane są identycznie jak tytuły rozdziałów za wyjątkiem wielkości czcionki – powinna wynosić ona 9/11 punktów. Inne są też odstępy (0.7 cm i 0.5 cm):

```
\SetSectionFormatting{subsection}
  {0.7cm}
  {\FormatBlockHeading{\centering%
    \secondarysize\MakeUppercase}}
  {0.5cm}
```

Bibliografia ma być składana czcionką mniejszej wielkości (`\secondarysize`) – i ten postulat realizuje klasa `mwart` – jako nienumerowany podrozdział. I to musi być zmodyfikowane. Aby zachować zgodność z resztą tekstu `\bibindent` ustawiamy na 0.5 cm.

Po bibliografii powinno pojawić się streszczenie w języku obcym (złożone podobnie jak streszczenie w języku polskim). Pewno można jakoś tak to zrobić, żeby w jednym miejscu definiować wszystkie streszczenia, a drukować je tam gdzie trzeba. Uznajmy jednak ten postulat, mimo, że słuszny i zgodny z zasadami „formatowania logicznego” [8] za wymyślny i zdefiniujmy kolejne otoczenie: `Abstract` (żeby nie przedefiniowywać istniejącego otoczenia `abstract`).

```
\newenvironment{Abstract}
  {\list{}{\leftmargin0.5cm%
```

```
\rightmargin0pt}\footnotesize%
\listparindent0.5cm%
\itemindent0.5cm\item\relax%
\centerline{\footnotesize ABSTRACT}%
\vskip5mm}
{\endlist}
```

## 6 I co dalej?

Udało się nam „zasymulować” wygląd narzucany przez instrukcję. Zapewne w konstrukcjach, które przedstawiłem, można jeszcze coś poprawić czy udoskonalić. Ale co z tego? Wszak wydawca (organizator konferencji) wymaga od nas pliku w formacie MS Word.

Jedyne sensowne rozwiązanie to skorzystanie z jakiegoś konwertera, który stworzy plik DOC w pełni odpowiadający stawianym wymaganiom. Ale problem polega na tym, że dobrych konwerterów tego typu nie ma.

Najdoskonalszym (darmowym) narzędziem jest `TeX4ht`. Zatem dalsza część czynności, które musimy wykonać to takie jego skonfigurowanie aby uzyskać efekt jak najbliższy oczekiwaniom.

Zwracam uwagę, że niedostarczenie przez Wydawcę zestawu stylów ułatwia nam nieco zadanie – możemy przygotować jakikolwiek styl spełniający wizualne wymagania. I generujący po stronie Wydawcy nowe problemy.

Uprzedzając fakty – proces konfigurowania `TeX4ht` nie jest bardzo prosty. Musi przebiegać zupełnie inaczej w przypadku konwertowania do Worda (konwersja robiona jest zestawu HTML/CSS), a zupełnie inaczej w przypadku OpenOffice.org (konwersja via XML/XSLT/MathML). Opis w `LATeX WEB Companion` [1] zawiera bardzo wiele przydatnych odpowiedzi, ale mocno nie odpowiada już wersji współczesnej (co zauważa Wisse [6]), jego wskazówki są również już częściowo nieaktualne kolejne 4 lata później. Pozostaje zatem analiza kodu i czytanie dokumentacji on-line uzyskanej z kompilacji pliku `mktex4ht.4ht`.

W dalszej części skupię się na zadaniu pierwszym (konwersja do MS Worda) mimo, że droga do Worda via OpenOffice.org wydaje się bardziej obiecująca. Niestety osobiste doświadczenia w tym zakresie już nie są tak dobre.<sup>1</sup>

## 7 TeX4ht

System `TeX4ht` [1, 2] to bardzo zaawansowane narzędzie konwersji (La)TeX → HTML. W odróżnieniu od

<sup>1</sup> Arytuł przygotowany w OO i lekko zmodyfikowany w Wordzie przestał się otwierać gdziekolwiek i został od nowa przepisany...

innych tego typu narzędzi wykorzystuje ono w maksymalnym stopniu  $\TeX$ a do interpretacji dostarczonego kodu źródłowego oraz szeregu plików konfiguracyjnych do wykonania konwersji.

Oprogramowanie zawarte w pakiecie TeXLive – instaluje się bardzo łatwo. Niestety rekonfiguracja zachowania pakietu nie musi być łatwa mimo, że dokumentacja [2] jest dosyć obszerna.

Autor pakietu dostarczył kilka gotowych skryptów służących do konwersji kodu TeXa/MeXa/LaTeXa/ConTeXa do (X)HTMLa, XMLa, DOCbooka, XML TEI, OpenOffice.org, MS Worda. Konwersje do formatów WYSIWYGowych nie przenoszą w pełni ani wyglądu ani struktury dokumentu. . .

Konwerter do OpenOffice.org znakomicie tłumaczy wzory matematyczne (via MathML), ale zupełnie ignoruje strukturę dokumentu (zachowując jedynie anglosaskie zasady wcięć akapitowych). Dobrą stroną takiego rozwiązania jest to, że użytkownik zachowuje pełną kontrolę nad formatowaniem tekstu. Opracowanie odpowiednich arkuszy stylu może być jednak interesującym wyzwaniem.

W przypadku konwersji do „formatu MSWord” (polecenie `wlatex`) wzory matematyczne zamieniane są na rastrowe pliki graficzne kiepskiej jakości. Wszystkie informacje o formatowaniu przenoszone są via arkusz CSS. I większość zmian w konfiguracji będzie musiała dotyczyć właśnie zmian w pliku CSS.

Konfiguracja  $\TeX$ 4ht nie jest sprawą prostą. Zaproponowana przez Autora pakietu metoda konwersji  $\LaTeX \rightarrow$  Word wiedzie przez HTML + CSS i w większości jest zrealizowana sensownie. Trzeba jednak dokonać kilku korekt związanych z zachowaniem wymagań **wizualnych** (realizowane będą za pomocą stylów). Program Word czytając kod HTML uwzględni wiele informacji zawartych w powiązanych z nim Kaskadowych Arkuszach Stylów. Jest to dobra wiadomość – gdyż mamy szansę na stworzenie w miarę sensownego pliku `l.doc`.

Większość modyfikacji będzie wymagało jedynie dodatków do pliku CSS – więc nie będzie to bardzo skomplikowane.

1. Wymiary strony i tak trzeba będzie ostatecznie ustawić w Wordzie (po zaimportowaniu pliku HTML). Dokonamy prostej modyfikacji szerokości żeby lepiej wyglądało w przeglądarce:

```
\Css{body {font-size: 11pt; text-align: justify; width: 13.5cm;}}
```

2. Standardowo, każdy ustęp tekstu poprzedzony jest niewielkim odstępem (HTMLowy odpowiednik `parskip`); musimy to zmienić:

```
\Css{p {margin-top: 0pt; margin-bottom: 0pt; font-size: 11pt; }}
```

```
\Css{p.indent{ text-indent: 0.5cm }}
```

Dodatkowo ustawiamy wielkość wcięcia akapitowego na 5 mm.

3. Tytułaria (section, subsection)

```
\Css{h3 {margin-top:1cm; margin-bottom: 0.7cm; text-transform: uppercase; text-align: center; font-size: 11pt;}}
```

Ustalamy odstępy przed i po tytule, zamianę na wersaliki, wymiar czcionki i centrowanie. Podobnie

```
\Css{h4 {margin-top:.7cm;margin-bottom: 0.5cm; text-transform: uppercase; text-align: center; font-size: 9pt;}}
```

Formatowanie logiczne to prawdziwa przyjemność!

4. Odpowiednie odstępy przed i po wzorach wystawionych powinna zapewnić następująca konstrukcja:

```
\Css{table.equation {margin-top:0cm; margin-bottom:0cm;}}
```

```
\Css{center.math-display {margin-top: 0.5cm; margin-bottom:0.5cm;}}
```

Niestety wymagało to dokładnego przyglądania się konstrukcji generowanego kodu HTML i plku CSS.

5. Podpisy pod rysunkami (i zawartość ilustracji oraz tabel) składamy czcionką wielkości 9 punktów:

```
\Css{.caption {font-size:9pt;}}
```

```
\Css{.float {font-size:9pt}}
```

Inne zmiany będą wymagały podjęcia bardziej skomplikowanych działań. Dotyczy to problemu różnego traktowania ustępów tekstu „normalnych” i tych występujących jako pierwsze po tytułariach.  $\TeX$ 4ht dodatkowo definiuje trzecią kategorię: tekst występujący zaraz po wzorze – traktowany jest on jak kolejny ustęp tekstu, który nie jest „wcinany”.

$\TeX$ 4ht pozwala na dosyć elastyczne konfigurowanie tłumaczenia różnych poleceń na kod wynikowy. Służy do tego polecenie `\Configure`. W przypadku tytułariów ma ono cztery parametry. Każdy z nich definiuje kod, który będzie wstawiony:

1. Przed zawartością jednostki tekstu.
2. Po zakończeniu.
3. Przed tytułem.
4. Po tytule.

A standardowa konfiguracja wygląda tak:

```
\Configure{section}{-}{-}
{\ifvmode \IgnorePar\fi \EndP\IgnorePar
```

```
\HCode{<h3 class="sectionHead"\a:L Rdir>}
\TitleMark\space\Htm1ParOff}
{\HCode{</h3>}\Htm1ParOn \IgnoreIndent
\par}
```

Pierwsze dwa parametry są puste. Trzeci parametr – bardzo istotny – jeżeli jest taka potrzeba najpierw kończy poprzedzający ustęp tekstu, a później definiuje odpowiednie znaczniki HTML (łącznie z deklaracją znacznika <a> pozwalającego na hipertekstowe odwołania do tytułu rozdziału. Następnie pojawia się sam tytuł i znacznik zamykający.

Wśród poleceń występuje `\IgnoreIndent` odpowiedzialne za brak wcięcia akapitowego w bezpośrednio następującym ustępie tekstu. Musimy to polecenie usunąć. Podobnie postępujemy z definicją polecenia `subsection`.

Właściwości nowodefiniowanych otoczeń ustala się poleceniem `\ConfigureEnv`:

```
\ConfigureEnv{streszczenie}
{}{}
{\ifvmode \IgnorePar\fi \EndP\HCode{<div
id="streszczenie"}}
```

```
{\ifvmode \IgnorePar\fi \EndP
\HCode{</div>}}
```

przerzucając formatowanie wizualne na Kaskadowy Arkusz Stylu:

```
\Css{#streszczenie {margin-left: 0.5cm;}}
```

Wszystkie zmiany konfiguracyjne `TeX4ht` zapisujemy w pliku o rozszerzeniu `.cfg` (w moim przypadku `alaword.cfg`). Żeby `TeX4ht` czytał ten plik konfiguracyjny należy wywoływać go podając nazwę pliku konfiguracyjnego jako drugi parametr:

```
wllatex przykład alaword
```

nie trzeba podawać rozszerzeń plików.

## 8 Podsumowanie

Przedstawione modyfikacje nie są niestety doskonałe. Wskazują jedną z kilku możliwych dróg podczas konwersji dzieła do nieprzyjaznego świata programów WYSIWYG.

Pokazują też, że niewielkie modyfikacje konfiguracji plus zabawa z arkuszami CSS pozwalają na uzyskanie, po zaimportowaniu do Worda, ostatecznego efektu zbliżonego do **wizualnych** wymagań Wydawcy. Nie można, oczywiście, w ten sposób wyeliminować całkowicie programu WYSIWYG. Sprawa poważnie się skomplikuje, gdy Wydawca przygotowuje arkusz stylu.

Pojawia się też pytanie o sensowność (czy ekonomię) przedstawionego postępowania. I tu odpowiedź jest oczywista (podobnie jak w przypadku każdego innego zadania do wykonania którego zatrudniamy

komputery): działania takie mają sens tylko i wyłącznie w tym przypadku gdy czas poświęcony na przygotowanie narzędzia (zaprogramowanie komputera) jest krótszy niż łączny czas używania narzędzia (w jakimś okresie).

Alternatywne podejście może polegać na opracowaniu (doskonalszych niż obecne) reguł do konwersji `LaTeX` → `OpenOffice.org`. I użycie tego programu do dalszej konwersji do Worda. Złożoność problemu jest zbliżona.

W przypadku konieczności przygotowania dokumentu, który później musi „żyć” w świecie biurowym – wykorzystuję gotowe konwertery dostarczane z systemem `TeX4ht`: `wllatex`, `htllatex` i `oolatex`. Jakość uzyskanych dokumentów jest wystarczająca.

## Literatura

- [1] M. Goossens, S. Rahtz. *The LaTeX Web Companion*. Addison-Wesley, 1999.
- [2] Eitan M. Gurari. `TeX4ht: LaTeX and TeX for Hypertext`. <http://www.cse.ohio-state.edu/~gurari/TeX4ht/mn.html>, 2005.
- [3] Instrukcja dla autorów artykułów. Dokument rodzaju prawa powielaczowego, 2005.
- [4] The LaTeX Project Public License. Wersja 1.3a dostępna jako <http://www.latex-project.org/lppl/>, 2006.
- [5] Peter Wilson. The `ccaption` package. 2005.
- [6] Maarten Wisse. Hacking `TeX4ht` for XML Output. *MAPS*, 28:28–35, 2002. [http://www.ntg.nl/maps/pdf/28\\_9.pdf](http://www.ntg.nl/maps/pdf/28_9.pdf).
- [7] Marcin Woliński. MWCLS Moje własne klasy dokumentów dla `LaTeX 2ε`. Podręcznik użytkownika. 2003.
- [8] Marcin Woliński. Adiustacja logiczna w praktyce. *Buletyn GUST*, 2006.