

UNIWERSYTET W BIAŁYMSTOKU

WYDZIAŁ MATEMATYCZNO-FIZYCZNY

INSTYTUT MATEMATYKI

Marzena Kropiewnicka

SKŁAD CZASOPISMA W L^AT_EX-U

*Praca dyplomowa napisana
pod kierunkiem
dr. Mariusza Żynela*

Białystok 2006

Składam serdeczne podziękowania
dr. Mariuszowi Żynelowi
za wsparcie i pomoc
w pisaniu pracy oraz
mojemu ojcu za to, że
we mnie wierzył.

Marzena Kropiewnicka

Spis treści

Wstęp	1
1 Prace matematyczne w \LaTeX-u	2
1.1 \LaTeX jako system składu dokumentów	2
1.2 Standardy składu artykułów matematycznych	3
1.3 Programowanie w \LaTeX -u	7
2 Zastosowanie Perla	10
2.1 Wyrażenia regularne	10
2.2 Obsługa plików	11
2.3 Moduł LaTeX::TOM	12
3 Realizacja czasopisma w \LaTeX-u	16
3.1 Jednolitość i spójność czasopisma	16
3.2 Klasa artykułu	17
3.3 Klasa zeszytu	22
3.4 Programy do obróbki zeszytu	24
A Klasa artykułu tga.cls	26
B Klasa zaszytu tgja.cls	36
C Program labels.cls	42
Bibliografia	43

Wstęp

Od czasu powstania systemu $\text{T}_{\text{E}}\text{X}$ niemal wszystkie szanujące się wydawnictwa używają go do składu prac matematycznych. Opracowanie nakładki $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ i powstanie wielu dodatkowych modułów rozszerzających jego funkcjonalność spowodowało, że stał się on bardzo popularny w środowisku naukowym. Jest on podstawowym narzędziem w warsztacie niemal każdego matematyka.

Mimo dużej elastyczności $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a, ze składem czasopism wiąże się wiele problemów. Biorą się one głównie stąd, że poszczególne artykuły wchodzące w jego skład są pisane przez różnych autorów, zgodnie z różnymi konwencjami i nawykami. Aby zadanie przygotowania zeszytu, pewnego wymyślanego czasopisma o nazwie *Traditional Geometry*, nieco ułatwić opracowaliśmy dwie klasy dokumentu w $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -u: *tga* oraz *tgj*. Pierwsza służy do składu pojedynczego artykułu przez samego autora, przed wysłaniem artykułu do publikacji. Druga klasa służy do składu całego zeszytu. Każdy artykuł wchodzący do zeszytu traktowany jest jak rozdział w książce. Naturalnym jest, że formatowanie takiego „rozdziału” w *tgj* musi być dokładnie takie samo jak formatowanie artykułu w *tga*. Zdecydowaliśmy w związku z tym, że klasa *tgj* dziedziczy po klasie *tga*, innymi słowy, klasa *tgj* jest nadbudową nad klasą *tga*.

Poza tym opracowaliśmy jeszcze skrypt w języku Perl, służący do porządkowania etykiet w artykułach, tak aby uniknąć powtarzania się ich w obrębie zeszytu.

Rozdział 1

Prace matematyczne w L^AT_EX-u

1.1 L^AT_EX jako system składu dokumentów

T_EX jest komputerowym systemem składu drukarskiego i łamania tekstów, stworzonym przez wybitnego matematyka i informatyka, Donalda E. Knutha w latach 1977-1982. Ponoć, rozczarowany niską jakością typograficzną swoich prac, Knuth zaprzestał publikowania i rozpoczął prace nad T_EX-em. Pierwsze wydanie systemu pojawiło się w roku 1982 i, co ciekawe, niemalże nie zmieniło się po dzień dzisiejszy. T_EX odniósł ogromny sukces, głównie w dziedzinie publikacji matematycznych. Okazało się jednak, że jest to system bardzo skomplikowany i korzystanie z niego stwarzało liczne problemy, szczególnie początkującym użytkownikom. Rozwiązaniem problemu okazał się L^AT_EX zbiór makrodefinicji ułatwiających korzystanie z T_EX-a.

T_EX umożliwia efektywne składanie tekstów o dowolnej trudności. Unikalny algorytm, którym posługuje się T_EX przy składaniu akapitów, powoduje, że nie ma programu oferującego w tym względzie lepsze możliwości. Inne zalety T_EX-a to jego cena i powszechność. Słowo T_EX oznacza także pewien wyspecjalizowany język programowania. Jak każdy język programowania, T_EX ma specyficzną składnię (zawiera m.in. zmienne, instrukcje podstawienia i instrukcje warunkowe) oraz pozwala definiować nowe funkcje.

L^AT_EX powstał w połowie lat osiemdziesiątych w laboratorium badawczym firmy SRI International i został szczegółowo opisany przez jego twórcę, Leslie Lamport, w książce [1]. Pierwowzorem był język Scribe. L^AT_EX to system składu posiadający ogromne możliwości. Między innymi są to: tworzenie wszelkiego rodzaju dokumentów technicznych, naukowych, listów, reportaży czy nawet książek. Praktycznie nadają się on do wszystkiego. Dokumenty L^AT_EX-a są plikami tekstowymi zawierającymi oprócz treści instrukcje formatujące. Pod tym względem jest to format bardzo przypominający dokumenty HTML. Praca w systemie L^AT_EX polega na:

- przygotowaniu edytorem tekstowym pliku źródłowego *tex*,
- przekompilowanie dokumentu *tex* programem L^AT_EX.

W wyniku otrzymujemy w zależności od konfiguracji oprogramowania dokumenty *dvi*, *ps* lub *pdf*. Zalety i wady L^AT_EX-a możemy oceniać na trzech płaszczyznach. Najważniejszą z nich jest jakość typograficzna otrzymywanych dokumentów. Drugą wkład pracy autora publikacji w poznanie L^AT_EX-a. Trzecią zaś, podatność na modyfikacje i dostosowanie do własnych potrzeb. Argumenty, które przemawiają za używaniem L^AT_EX-a, to:

- dostępność gotowych, przygotowanych przez zawodowców układów graficznych,
- wygodne składanie wzorów matematycznych,
- istnienie wielu bezpłatnych pakietów poszerzających możliwości typograficzne L^AT_EX-a,
- używając L^AT_EX-a autor skupiony jest na treści nie na wyglądzie,
- w L^AT_EX-u opisuje się logiczną strukturę dokumentu.

1.2 Standardy składu artykułów matematycznych

Klasa dokumentu jest to szablon określający jego wygląd. Są w niej zawarte informacje na temat użytych czcionek, wcięć, odstępów, odległości, jak również algorytmy pomagające decydować jak program ma się zachować w rozmaitych sytuacjach. Zadeklarowanie klasy jest pierwszą rzeczą, którą należy zrobić po przystąpieniu do wykonywania składu. Deklaracja ta musi wystąpić w pierwszej linii pliku. Deklaracja klasy ma postać:

```
\documentclass [parametry] {klasa}
```

Powyższe polecenie otwiera tzw. *preambulę* dokumentu, w której zamieszcza się polecenia wczytujące pakiety rozszerzeń, dodatkowe deklaracje, własne makrodefinicje i inne elementy sterujące wyglądem. L^AT_EX zawiera cztery podstawowe klasy dokumentów:

- klasa *article* - służąca do składu artykułów, sprawozdań i innych krótkich dokumentów,
- klasa *report* - służąca do wykorzystania w raportach i dłuższych opracowaniach,
- klasa *book* - mająca zastosowanie do składu książek i innych materiałów o dużej objętości,
- klasa *letter* - stanowiąca szablon listu.

Do klas dokumentów obsługiwanych przez L^AT_EX-a zalicza się również klasę *slide*, służącą do składu prezentacji przeznaczonych do wydrukowania na folii i projekcji przy użyciu rzutnika. Przy doborze klasy nie należy sugerować się ich nazwami, a jedynie tym co ma zawierać przygotowywany dokument. Klasa dokumentu jest rzeczą umowną i tylko od nas zależy którą wybierzemy. Oczywiście każda z nich zawiera swoje cechy charakterystyczne, upodabniające ją do elementu, do którego miała w założeniu służyć.

Parametry podawane w wywołaniu klasy precyzują istotne elementy wyglądu dokumentu. Najważniejsze z nich to:

draft Wymusza przetwarzanie dokumentu w trybie roboczym; tryb ten powoduje między innymi zaznaczanie na marginesach miejsc występowania zbyt szerokich elementów powodujących błąd *Overfull badness...* oraz wyłącza interpretację wstawionych rysunków, oznaczając je prostokątami o zadeklarowanym rozmiarze.

final Wymusza przetwarzanie dokumentu na gotowo do druku.

10pt Ustawia stopień pisma tekstu głównego dokumentu na 10 punktów.

11pt Ustawia stopień pisma tekstu głównego dokumentu na 11 punktów.

12pt Ustawia stopień pisma tekstu głównego dokumentu na 12 punktów.

a4paper Ustawia rozmiar papieru na format A4.

letterpaper Ustawia rozmiar papieru na format letter.

landscape Powoduje zmianę orientacji strony na poziomą (właściwie zamienia ona tylko szerokość papieru z jego wysokością, natomiast nie powoduje obrócenia zawartości strony).

fleqn Powoduje dosunięcie równań do lewej strony.

leqno Wymusza umieszczanie numeracji równań po ich lewej stronie.

titlepage Wymusza złamanie strony po tytule.

notitlepage Zabrania złamania strony po tytule.

onecolumn Wymusza skład w jednym łamie.

twocolumn Wymusza skład w dwóch łamach.

oneside Wymusza skład jednostronny.

twoside Wymusza skład dwustronny.

openright Wymusza rozpoczęcie nowego rozdziału na stronie nieparzystej (prawej).

openany Zezwala na rozpoczęcie nowego rozdziału na stronie nieparzystej lub parzystej.

openbib Powoduje zmianę sposobu składu bibliografii. Pierwsza linia pozycji będzie dosunięta do lewego marginesu, natomiast następne zostaną wcięte o wartość podaną parametrem `bibindent`.

Do składu naszego czasopisma potrzebne są dwie klasy. Zeszyt składany będzie w sposób podobny jak składa się książkę w L^AT_EX-u. Rozdziały będą osobnymi artykułami. Aby zainteresowani autorzy mogli składać swoje artykuły zgodnie z formatem naszego czasopisma warto udostępnić im odpowiednią klasę artykułu. Oczywiście obie klasy będą miały wiele wspólnego, wręcz klasa zeszytu będzie zawierać klasę artykułu.

Istnieje wiele typowych sytuacji, które nie zostały przewidziane przez autora L^AT_EX-a, a co za tym idzie nie zostały w nim zaimplementowane. Jednak dzięki nieograniczonemu dostępowi do kodów źródłowych oraz liberalnej licencji programu możliwe jest definiowanie własnych makr rozszerzających dostępny zbiór poleceń. Takie zbiory makr nazywane są pakietami (`packages`).

Jednym z pakietów, istotnym z punktu widzenia prac matematycznych, pozwalającym na definiowanie numerowanych środowisk służących do składu twierdzeń, zadań, definicji itp. jest `amsthm`. Podstawowym poleceniem tego pakietu jest:

```
\newtheorem{nazwa}{tekst nagłówka}
```

Po wykonaniu takiego makra będzie można używać nowe środowisko o nazwie `nazwa` z podanym napisem w nagłówku. Postać wywołania zdefiniowanego środowiska jest następująca:

```
\begin{nazwa}
  zawartość środowiska
\end{nazwa}
```

lub

```
\begin{nazwa}[dodatek]
  zawartość środowiska
\end{nazwa}
```

gdzie `dodatek` może stanowić dodatkowy komentarz będący częścią nagłówka. Makro `\newtheorem` może pobierać argument opcjonalny na jeden z dwóch sposobów. Domyślnie numeracja definiowanych środowisk jest niezależna, tzn. osobno będą numerowane twierdzenia, osobno lematy itd. Jeżeli chcemy zastosować wspólny licznik, to przekazujemy go jako opcjonalny argument zgodnie z poniższym wzorcem:

```
\newtheorem{nazwa}[nazwa_innego_środowiska]{tekst nagłówka}
```


W tym wypadku licznikiem kolejnych pozycji będzie licznik określony przez `nazwa_innego_środowiska`. Tak więc wykonanie:

```
\newtheorem{thm}{Theorem}
\newtheorem{lem}[thm]{Lemma}
```

spowoduje, że lematy i twierdzenia będą miały wspólny licznik.

Zdefiniowanie środowiska w sposób jak poniżej:

```
\newtheorem{nazwa}{tekst nagłówka}[licznik_nadrzędny]
```

spowoduje, że licznik zliczający kolejne użycie środowiska `nazwa` jest zerowany przy zmianie licznika `licznik_nadrzędny` oraz w składzie numeru definiowanego środowiska będzie wchodził licznik nadrzędny. Licznikiem nadrzędnym może być `chapter`, `section`, `subsection` lub licznik innego środowiska zdefiniowanego za pomocą polecenia `\newtheorem`. Oba warianty dodatkowe wołania `\newtheorem` są wzajemnie wykluczające się.

Polecenie

```
\newtheorem*{nazwa}{tekst nagłówka}
```

powoduje, że środowisko nie będzie numerowane.

Definicja nowego środowiska może wyglądać następująco:

```
\newtheorem{thm}{Theorem}
\newtheorem{lem}[thm]{Lemma}
```

natomiast użycie jak poniżej:

```
\begin{thm}
  Treść twierdzenia.
\end{thm}
```

```
\begin{lem}
  Treść lematu.
\end{lem}
```

```
\begin{lem}
  Treść drugiego lematu.
\end{lem}
```

```
\begin{thm}[specjalne]
  Treść specjalnego twierdzenia.
\end{thm}
```

Sposób prezentacji środowiska jest zależny od wybranego stylu. Polecenie `\newtheorem` można poprzedzić poleceniem `\theoremstyle`. Domyślnie używany jest styl *plain*.

Wywołanie polecenie wygląda następująco: `\theoremstyle{style}` gdzie *style* może przyjmować następujące wartości:

plain emuluje oryginalny styl L^AT_EX-a (tekst nagłówka jest składany czcionką wytłuszczoną, a wewnątrz środowiska kursywą (bez przejścia do nowej linii po nagłówku),

definition jak powyżej, ale wewnątrz jest składane prostą czcionką,

remark tekst nagłówka składany jest kursywą (bez przejścia do nowej linii po nagłówku), a wewnątrz środowiska prostą czcionką.

Nie ma powszechnego standardu określającego jak mają się nazywać środowiska do składu twierdzeń, lematów, uwag, definicji itp. Przygotowując klasę artykułu matematycznego powinniśmy jednak na coś się zdecydować. W samym pakiecie *amsthm* jest sugestia jak takie środowiska nazywać. Ponadto wiele poważnych wydawnictw jak np. Elsevier Science stosuje się do tej sugestii. Nie pozostaje nam nic innego jak pozostać przy tym niepisanym standardzie. Dokładnie jakie środowiska wprowadzamy, zostanie opisane w podrozdziale 3.2.

Kolejnym pakietem, o którym warto wspomnieć mówiąc o składzie artykułów matematycznych jest pakiet *amsmath*. Pakiet ten stanowi zestaw makr wspomagających skład zaawansowanych tekstów matematycznych. Wśród tych makr znajdują się makra do składu rozbudowanych formuł matematycznych zajmujących kilka wierszy tekstu, macierzy z różnymi wariantami nawiasów, diagramów przemiennych, akcentów i strzałek dowolnej długości, zagnieżdżonych ułamków oraz symboli pochodnych od symbolu Newtona. Dzięki pakietowi *amsmath* łatwo można deklarować operatory takie jak np. `dim` i w jednoznaczny sposób odwoływać się do formuł matematycznych poleceniem `\eqref`.

1.3 Programowanie w L^AT_EX-u

Tekst właściwy każdego dokumentu zaczyna się od polecenia

```
\begin{document}
```

a kończy poleceniem

```
\end{document}
```

Fragment pliku źródłowego poprzedzony poleceniem `\documentclass` jest wspomnianą wcześniej *preambułą*. Jak pisaliśmy wcześniej, *preambuła* zaczyna się od polecenia `\documentclass`, którego argument jest nazwą jednej z predefiniowanych klas dokumentu. Polecenie `\documentclass` może być użyte z opcjami lub bez nich.

Klasa dokumentu zawiera definicje poleceń określających standardowe struktury tekstu L^AT_EX-a. Dodatkowe konstrukcje są zdefiniowane przez pakiety,

ładowane poleceniem `\usepackage`. Pakiet może mieć opcje podawane jako argument opcjonalny (opcje ujęte w nawiasy kwadratowe) polecenia `\uspackage`.

Najprostszym rodzajem powtarzalnej konstrukcji jest ten sam tekst występujący w różnych miejscach. Deklaracja `\newcommand` definiuje nowe polecenie, które ma generować ten tekst, jej pierwszym argumentem jest nazwa polecenia, a drugim tekst. Za pomocą polecenia `\newcommand` można zdefiniować skróty do poleceń, z których korzysta się najczęściej. Do przedefiniowania istniejącego już polecenia służy deklaracja `\renewcommand`, ma ona takie same argumenty jak `\newcommand`.

Do zdefiniowania nowego otoczenia służy polecenie `\newenvironment`. Natomiast, aby zdefiniować na nowo już istniejące środowisko, należy użyć polecenia `\renewenvironment`.

Polecenie `\input` umożliwia najprostszy podział tekstu źródłowego na kilka plików. Oprócz tego, że dzieli tekst źródłowy na poręczne kawałki, ułatwia użycie tego samego tekstu źródłowego w różnych dokumentach. Jeszcze jednym powodem dzielenia tekstu na oddzielne pliki jest uruchamianie L^AT_EX-a tylko na części dokumentu tak, aby po wprowadzeniu zmian były przetwarzane tylko zmienione fragmenty. Do tego celu służy polecenie `\include`. Za pomocą tego polecenia można kazać L^AT_EX-owi, albo wstawić plik, albo go pominąć i przetwarzać cały dalszy tekst, jakby tamten plik był wstawiony z numeracją stron, podrozdziałów, równań itp. taką, jak gdyby tekst pominiętego pliku był włączony.

L^AT_EX udostępnia wiele parametrów długości, które są parametrami składu tekstu. Parametr długości to wielkość liczbowa wraz z jednostką. Dopuszczalne jednostki to między innymi: *in*, *cm*, *mm*, *pt*, *ex* (wysokość małej litery x) oraz *em* (szerokość małej litery m). Każda konstrukcja w L^AT_EX-u, jak strona, lista, paragraf itp., posiada własny zestaw takich parametrów. Nie sposób opisać tutaj wszystkie parametry, nawet te standardowe ze względu na ich olbrzymią ilość. Ograniczymy się tylko do najbardziej podstawowych.

Za skład strony odpowiedzialne są następujące parametry:

topmargin – szerokość górnego marginesu,

headheight – wysokość nagłówka strony,

headsep – odstęp pomiędzy nagłówkiem, a tekstem na stronie,

textheight – wysokość tekstu na stronie,

footskip – odległość od dolnej krawędzi tekstu do dolnej krawędzi stopki,

oddsidemargin – szerokość marginesu na stronie nieparzystej,

evensidemargin – szerokość marginesu na stronie parzystej,

textwidth – szerokość tekstu na stronie.

Przy określaniu szerokości marginesów należy pamiętać, że L^AT_EX automatycznie zwiększa górny i lewy margines o 1 cal. Tak więc, aby uzyskać mniejszy margines, należy zastosować wartość ujemną.

O składzie akapitu decydują poniższe wielkości:

parindent – wielkość wcięcia na początku zwykłego akapitu,

parskip – dodatkowy pionowy odstęp ustawiony między akapitami,

baselineskip – normalny pionowy odstęp od dołu jednego wiersza do dołu następnego w tym samym akapicie.

Do wprowadzania nowych parametrów długości w L^AT_EX-u służy makro `\newlength`, które jest nadbudową nad T_EX-owym `\newdimen`. Przy pomocy makra `\setlength` ustawia się parametr długości na zadaną wielkość, np.

```
\setlength{parskip}{1mm}
```

ustawia wielkość `parskip` na 1mm. Makro `\addtolength` zwiększa wartość parametru o zadaną jako argument wielkość. Oczywiście możliwe jest podanie wielkości ujemnej.

Rozdział 2

Zastosowanie Perla

Nie wszystkie zadania związane ze składem czasopisma w \LaTeX -u można rozwiązać na poziomie samego \LaTeX -a. Na przykład problem taki, jak prefiksowanie etykiet, czy wyszukanie pakietów w artykule, który ma być umieszczony w zeszytcie, może być stosunkowo łatwo rozwiązany przy pomocy zewnętrznego narzędzia. Ze względu na szerokie zastosowanie i duże możliwości, świetnie nadaje się Perl.

Perl to z założenia praktyczny, interpretowany język programowania przeznaczony przede wszystkim do wykonywania różnych zadań systemowych m.in. przeszukiwania plików tekstowych, pobierania z nich informacji i generowania komunikatów na ich podstawie. Świetnie nadaje się do pisania skryptów CGI umieszczanych w witrynach WWW. Jego autorem jest Larry Wall wspomagany oczywiście przez ogromną liczbę innych programistów. Perl jest kompletny, stosunkowo łatwy w użyciu oraz wydajny, przypłacając to może nieco elegancją. Posiada wiele cech języka C, jak również sed, awk i sh. Perl został zaprojektowany jako praktyczne narzędzie do analizy plików tekstowych i tworzenia raportów. Wszechstronność Perla pozwala na programowanie w różnych modelach: proceduralnym, funkcyjnym czy obiektowym.

2.1 Wyrażenia regularne

Wyrażenia regularne są nieodłączną częścią większości języków programowania, edytorów i narzędzi programistycznych. Stanowią wyjątkowo skuteczny mechanizm przetwarzania wszelkich danych, w tym tekstów, np. poczty elektronicznej. Od wielu lat występują również we wszystkich sytuacjach związanych z przeszukiwaniem i przetwarzaniem tekstu. Wyrażenie regularne to sposób na opisanie jednym, krótkim wzorcem zarówno prostych ciągów znaków, jak i złożonych układów.

Poniższy przykład ma za zadanie czytać dane ze standardowego wejścia (STDIN), wiersz po wierszu, szukać w nich słowa `czasopismo` i wypisywać wiersze, w których znajdzie to słowo, na standardowe wyjście (STDOUT):

```
#!/usr/bin/perl
while ( $wiersz = <STDIN> ) {
    print $wiersz if $wiersz =~ /czasopismo/;
}
```

Kluczowym fragmentem jest tutaj warunek `$wiersz =~ /czasopismo/`. Zawiera on dwa istotne elementy. Po pierwsze, samo wyrażenie regularne zostało zapisane w ukośnikach, po drugie zostało porównane ze zmienną `$wiersz` nie przy pomocy tradycyjnego porównania, ale z użyciem operatora `=~`, który powoduje, że jego prawa strona traktowana jest jako wyrażenie regularne. Ważne są wielkie i małe litery. Jeżeli program ma rozpoznawać zarówno słowo `Czasopismo` jak i `czasopismo`, należy go nieco zmodyfikować i warunek zapisać następująco: `$wiersz =~ /[Cc]zasopismo/`. Nawiasy kwadratowe oznaczają „jeden z ...”. Można używać tutaj także zakresów. Przykładowo `[0-9]` odpowiada zapisowi `[0123456789]`, czyli dowolnej cyfrze. Możliwe jest również stosowanie zaprzeczenia. Zapis `[^0-9]` oznacza znak, nie będący cyfrą. Innym skrótem, jaki można zastosować, jest kropka. Oznacza ona dowolny znak. Inne dwa ważne specjalne symbole w konstruowaniu wyrażeń regularnych określają jak wiele razy może wystąpić dany znak lub grupa znaków. I tak `*` mówi, że poprzedzający ją znak lub grupa może powtórzyć się 0 lub więcej razy, natomiast `+` mówi, że to co go poprzedza może powtórzyć się 1 lub więcej razy.

Wyrażenia regularne mogą być przydatne nie tylko przy wyszukiwaniu, ale także przy łatwej zamianie odszukanego tekstu na inny. Poniższy kod zamienia wszystkie pierwsze (w każdym wierszu) wystąpienia wyrazu `czasopismo` na `artykuł`.

```
#!/usr/bin/perl
while ( $wiersz = <STDIN> ) {
    $wiersz =~ s/czasopismo/artykuł/;
    print $wiersz;
}
```

Nie wyczerpiemy w tym miejscu tematu wyrażeń regularnych (por. [5, 6]), chcemy go tylko zasygnalizować, jako mocne narzędzie, które stosujemy w rozwiązywaniu problemów związanych z realizacją tej pracy.

2.2 Obsługa plików

Każdy plik przed zapisem lub odczytem należy otworzyć, a po zakończeniu operacji na nim - zamknąć. Do otwarcia jakiegokolwiek pliku w języku Perl służy rozbudowana funkcja `open`, zaś do zamknięcia - `close`. Nie ma żadnych limitów ilości otwieranych plików w Perlu poza zewnętrznymi, systemowymi. Schemat obsługi pliku jest prosty:

```
open(DANE, ścieżka_do_pliku); # otwarcie pliku
```

```
[ ... ]           # operacje na pliku
close(DANE);      # zamknięcie pliku
```

Zmienna DANE jest skojarzona z plikiem do czasu jego zamknięcia. Przy jej pomocy odwołujemy się do tego pliku.

Najprostszym przypadkiem wprowadzania danych z pliku będzie wczytanie do tablicy całej zawartości pliku tekstowego:

```
open(DANE, '/etc/passwd'); # otwarcie pliku do odczytu
@tekst <DANE>;             # zapamiętanie tekstu z pliku w tablicy
close(DANE);               # zamknięcie pliku
```

W powyższym przykładzie czytany jest cały plik jednym poleceniem. Kolejne wiersze tego pliku będą kolejnymi wyrazami w tablicy @tekst. Można też czytać pliki wierszami:

```
open(DANE, '/etc/passwd'); # otwarcie pliku do odczytu
while (<DANE>) {           # czytanie pliku wiersz po wierszu
    print $_ if (!/^%/);   # wypisanie wiersza
}
close(DANE);               # zamknięcie pliku
```

Przy czytaniu pliku wiersz po wierszu w zmiennej \$_ umieszczany jest bieżąco przetwarzany wiersz. W powyższym przykładzie wypisane zostaną tylko te wiersze pliku, które nie zaczynają się znakiem %.

Dane do pliku zapisujemy w poniższy sposób:

```
open(DANE, '>dane.txt');   # otwarcie pliku do zapisu
print DANE "Zapisywany tekst.\n" # zapisanie tekstu do pliku
close(DANE);               # zamknięcie pliku
```

2.3 Moduł LaTeX::TOM

W zasadzie jedynym gotowym do użycia parserem do L^AT_EX-a w Perlu jest moduł LaTeX::TOM. Nazwa TOM jest skrótem od TeX Object Model. Moduł ten został zaprojektowany do przetwarzania plików L^AT_EX-owych w nastawieniu na ekstrakcję zwykłego tekstu i jego modyfikację.

Moduł LaTeX::TOM dostarcza parser, który skanuje i interpretuje, chociaż nie w pełni dokumenty L^AT_EX-owe. Parser ten jako wynik skanowania zwraca reprezentację dokumentu w postaci drzewa. To drzewo jest obiektem o nazwie LaTeX::TOM::Tree.

Moduł LaTeX::TOM rozpoznaje pięć konstrukcji logicznych L^AT_EX-a. Stąd pięć możliwych typów węzłów w drzewie LaTeX::TOM::Tree. Są to:

TEXT – Ten typ węzła reprezentuje fragmenty zwykłego tekstu zawartego w dokumencie. Zawiera on wzory matematyczne oraz wszystko, co nie zostanie rozpoznane jako makro.

```
\label{etykieta}
```

Zostanie ono zidentyfikowane jako węzeł COMMAND, który będzie posiadał jednoelementowe poddrzewo posiadające węzeł typu TEXT, zawierający napis `etykieta`.

ENVIRONMENT – Otoczenia (środowiska) reprezentowane są jako węzły typu ENVIRONMENT. Zawierają one metadane na temat danego otoczenia oraz poddrzewo reprezentujące to, co zawarte jest w tym otoczeniu. Na przykład środowisko:

```
\begin{equation}
  r \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

zostanie zeskanowane jako węzeł ENVIRONMENT z nazwą `equation`. Związane z nim poddrzewo zawierać będzie rezultat skanowania powyższej formuły matematycznej.

GROUP – Obiekt tego typu jest poddrzewem przedstawiającym efekt skanowania tego, co zostało umieszczone w nawiasach klamrowych (`{}`). W nawiasach tych umieszcza się fragmenty kodu L^AT_EX-owego, aby semantycznie odizolować je od reszty kodu. Jest to zachowywane przez parser `LaTeX::TOM`.

COMMENT – Węzeł COMMENT jest bardzo podobny do węzła typu TEXT. Różnica polega na tym, że zawiera on fragmenty tekstu poprzedzone znakiem `%`.

Moduł `LaTeX::TOM` dostarcza trzy obiekty, które teraz opiszemy. Skupimy się wyłącznie na metodach i własnościach, które wykorzystujemy w pracy.

LaTeX::TOM::Parser

Poza konstruktorem tego obiektu dostępne są dwie metody:

`parseFile(filename)` – Metoda ta czyta plik o nazwie `filename` i skanuje go zwracając obiekt `LaTeX::TOM::Tree`.

`parse(string)` – Skanuje napis `string` i zwraca jako wynik obiekt `LaTeX::TOM::Tree`.

LaTeX::TOM::Tree

Opisywany tu obiekt reprezentuje strukturę dokumentu w postaci drzewa. Między innymi posiada następujące metody:

`print` – Wypisuje pełną strukturę drzewa w postaci tekstowej.

`toLaTeX` – Zwraca napis w języku L^AT_EX odpowiadający dla danego drzewa. Szczególnie przydatna metoda, gdy chcemy zapisać zmiany dokonane w dokumencie wejściowym.

`getCommandNodesByName(name)` – Zwraca tablicę adresów wszystkich węzłów typu COMMAND o nazwie `name`.

`getEnvironmentsByName(name)` – Zwraca tablicę adresów wszystkich węzłów typu ENVIRONMENT o nazwie `name`.

`getNodeByCondition(expression)` – Zwraca tablicę adresów wszystkich węzłów, które spełniają podane wyrażenie `expression`. Jest to bardzo wygodna metoda do przeszukania całego drzewa dokumentu i wyłapania interesujących nas fragmentów. Wyrażenie `expression` jest dowolnym, logicznym wyrażeniem w Perlu, co świadczy o dużych możliwościach tej metody. W wyrażeniu to odwołuje się do przetwarzanego węzła za pomocą zmiennej `$node`.

LaTeX::TOM::Node

Obiekt ten odzwierciedla poszczególne konstrukcje dokumentu L^AT_EX-owego zwane węzłami.

`getNodeType` – Zwraca typ węzła, czyli jedno z TEXT, COMMAND, ENVIRONMENT, GROUP, albo COMMENT.

`getNodeText` – Ta metoda może być użyta w kontekście TEXT albo COMMENT i zwraca tekst zawarty w odpowiednim węźle,

`setNodeText(text)` – Ustawia wartość węzłów tekstowych TEXT i COMMENT na podaną wartość `text`.

`getEnvironmentClass` – Dla otoczeń, czyli węzłów typu ENVIRONMENT zwraca nazwę otoczenia.

`getCommandName` – Zwraca nazwę makra. Ma oczywiście sens wyłącznie dla węzłów typu COMMAND.

`getChildTree` – Zwraca obiekt LaTeX::TOM::Tree będący drzewem węzłów „poniżej” danego węzła. Stosuje się wyłącznie do węzłów typu COMMAND, ENVIRONMENT lub GROUP.

`getFirstChild` – Zwraca obiekt `LaTeX::TOM::Node` będący pierwszym węzłem poddrzewa obiektów poniżej danego węzła. Metoda ta może być użyta dla węzłów typu `COMMAND`, `ENVIRONMENT` lub `GROUP`.

`getPreviousSibling` – Zwraca adres węzła znajdującego się bezpośrednio przed danym węzłem, na tym samym poziomie drzewa.

`getNextSibling` – Zwraca adres węzła znajdującego się bezpośrednio za danym węzłem, na tym samym poziomie drzewa.

`getParent` – Zwraca adres węzła bezpośrednio nad danym węzłem.

Rozdział 3

Realizacja czasopisma w L^AT_EX-u

3.1 Jednolitość i spójność czasopisma

Tak jak pisaliśmy we wstępie nasze czasopismo to zbiór artykułów matematycznych nadsyłanych przez wielu różnych autorów. Ze złożeniem tych odrębnych artykułów w jeden jednolity i spójny zeszyt wiąże się wiele technicznych problemów.

Podstawową kwestią przy składzie zeszytu jest zadbanie o jednolity wygląd poszczególnych artykułów. W tym celu zaprojektowaliśmy dwie klasy, jedną do składu artykułów przez samych autorów, drugą do składu zeszytów czasopisma.

Poważnym problemem przy składzie czasopisma jest unikalność prywatnych identyfikatorów wymyślonych przez autorów. Mamy tu na myśli etykiety oraz nazwy makr. Różni autorzy, w różnych artykułach, mogą użyć jednakowych identyfikatorów. W przypadku etykiet spowoduje to błędy w powoływaniach nie tylko przy podpisach do rysunków, czy tabel, ale w dowodach twierdzeń. Błędy mogą też wystąpić przy powoływaniach na literaturę. Jeśli chodzi o prywatne makra problem można obejść wykonując każdy z artykułów w odizolowanym otoczeniu. Można to uzyskać stosując makra `\begingroup` oraz zamykające `\endgroup`. Sposób ten nie pomoże w przypadku etykiet, gdyż są one spisywane w pliku pomocniczym `aux` jako definicje etykiet (za pomocą makra `\newlabel`) i mogą się one przysłać. Słusznym rozwiązaniem wydaje się tutaj prefiksowanie wszelkiego rodzaju etykiet w artykułach, najlepiej identyfikatorem artykułu przyznanym podczas jego przyjmowania do publikacji.

Do prywatnych identyfikatorów należy też zaliczyć nazwy plików. Nazwę pliku każdego artykułu najlepiej zastąpić jego identyfikatorem. Należy tutaj założyć, że jednemu artykułowi odpowiada dokładnie jeden plik z tekstem L^AT_EX-owym. Do tego dochodzą pliki ewentualnych rysunków zawartych w artykule. Nazwy tych plików najlepiej zestandaryzować stosując poniższą maskę:

identyfikator_artukułu-fig-NN.eps

Zakładamy tutaj jednocześnie, że wszystkie rysunki zostaną przekonwertowane do formatu Postscript (**eps**).

Poważnym problemem podczas składu zeszytu są używane przez autorów dodatkowe pakiety makr. Po pierwsze musimy zdiagnozować jakie dodatkowe pakiety muszą być dodane do zeszytu tak, aby wszystkie zawarte w nim artykuły zostały prawidłowo złożone. W tym celu każdy z artykułów musi zostać rozdzielony na dwie części: preambułę i tekst właściwy. W preambule należy wyszukać wszystkie użycia makra `\usepackage`. Należy przy tym zwrócić uwagę na parametry opcjonalne przekazywane do pakietów. Poza `\usepackage` musimy też sprawdzić, czy nie są używane makra `\input` i `\include`, gdyż pakiety mogą być włączane również przy ich pomocy.

Drugi, dużo ważniejszy problem, to kompatybilność tych pakietów pomiędzy sobą oraz między nimi i naszą klasą. Niemożliwe jest niestety sprawdzenie kompatybilności bez zawołania programu L^AT_EX-a i obejrzeniu wyników. Jeśli zostaną wyłapane niekompatybilności to warto je spisywać. Najlepiej utworzyć coś w rodzaju prostej bazy danych niekompatybilności (czarnej listy) tak, aby nie powtarzać roboty w przyszłości.

3.2 Klasa artykułu

Przy opracowywaniu klasy *tga* wzorowaliśmy się na standardowych klasach *article* oraz *book*. Część konstrukcji jest z nich zaczerpnięta bez zmian lub tylko z kosmetycznymi zmianami. W dalszej części opisujemy dokładnie klasę *tga*, której kod znajduje się w Dodatku A na stronie 26.

Opcje klasy

Ponieważ zależy nam, aby autor mógł złożyć swoją pracę w formie dokładnie takiej samej jak zostanie to wykonane przy składzie całego numeru czasopiśma, nasza klasa nie zawiera opcji, którymi można zmieniać wygląd artykułu. Zrezygnowaliśmy tutaj z opcji dostępnych w standardowych klasach *article* i *book*, takich jak: rozmiar papieru, wielkość czcionki, tryb landscape, czy skład jednostronny. Dostępne są tylko dwie, wykluczające się wzajemnie opcje: *draft* i *final*. Mają one typowe znaczenie takie jak opisane wcześniej w podrozdziale 1.2. Jeśli, żadna z nich nie zostanie użyta zakłada się, że domyślną opcją jest *final*.

Wymagane pakiety

Podczas składu artykułu korzystamy z kilku typowych dla prac matematycznych pakietów, takich jak:

- *latexsym*,

- *amsmath*,
- *amsthm*,
- *amssymb*,
- *amsfonts*.

Są one dostępne w podstawowej dystrybucji L^AT_EX-a, nie powinno więc być problemów podczas składu artykułu. Poza opisanymi w podrozdziale 1.2 pakietami *amsmath* i *amsthm* pozostałe, jak sugerują ich nazwy, służą do podłączenia dodatkowych czcionek i symboli matematycznych.

Dane numeru czasopisma

Jest kilka parametrów globalnych stałych związanych z danym zeszytem czasopisma. Są też parametry na stałe przypisane czasopismu takie jak: nazwa czasopisma, skrót tej nazwy, numer ISSN czy właściciel praw autorskich. Dlatego też zdefiniowaliśmy odpowiednio makra `\tgname`, `\tgshortname`, `\issn` oraz `\copyrightholder`. Zostały one wprowadzone, aby ułatwić modyfikacje samej klasy, ponieważ nie wszystkie wartości są znane w momencie jej opracowywania.

Jeśli chodzi o parametry związane z konkretnym numerem czasopisma to wprowadziliśmy następujące makra: `\tg@year`, `\tg@volume`, `\tg@number`. Dodaliśmy do nich makra definiujące (bez znaku `@` w nazwie) oraz makra formatujące (z prefiksem `\the`). Wartości tych makr nie mają sensu poza konkretnym zeszytem, czyli w sytuacji, gdy z klasy korzysta autor artykułu. Dopiero w momencie publikacji artykułu są one znane. Makra te zostały tutaj wprowadzone z uwagi na zachowanie kompatybilności z klasą zeszytu *tgj*.

Deklaracje czcionek

Na potrzeby naszego czasopisma została zdefiniowana nowa rodzina czcionek o nazwie `\tgfamily`. Rodzina ta oparta jest o czcionkę Adobe Avant Garde Gothic. Dla wygody zostało również wprowadzone makro `\texttg` służące do wypisania podanego tekstu, przekazanego jako argument makra, przy użyciu tej specjalnej czcionki.

Parametry układu strony

Dwa najważniejsze parametry decydujące o układzie strony to rozmiar papieru. Wysokość i szerokość kartki określona jest w L^AT_EX-u za pomocą parametrów `paperheight` i `paperwidth`. Założyliśmy, że nasze czasopismo będzie drukowane na papierze formatu B5 i odpowiednio ustawiliśmy te parametry.

Ponadto, aby móc łatwo modyfikować układ strony artykułu wprowadziliśmy kilka naszych prywatnych parametrów o sugestywnych nazwach i zebrałiśmy je w jednym miejscu. Są to:

`p@binding` – szerokość grzbietu zeszytu,
`p@nonprintablemargin` – szerokość marginesu poza zasięgiem drukarki,
`p@horizontalmargins` – część strony przeznaczona na marginesy w poziomie,
`p@verticalmargins` – część strony przeznaczona na marginesy w pionie,
`p@marginbalance` – procentowa różnica szerokości marginesów lewego i prawego, wartość 0.5 oznacza, że margines lewy i prawy są równe,
`p@abovetitleskip` – odstęp przed tytułem artykułu,
`p@aboveauthorskip` – odstęp przed listą autorów,
`p@belowauthorskip` – odstęp poniżej listy autorów,
`p@normalheadheight` – wysokość nagłówka,
`p@normalheadsep` – odstęp pomiędzy nagłówkiem i tekstem,
`p@normalfootskip` – odstęp pomiędzy tekstem a linią bazową stopki.

Układ strony

Aby łatwo móc zmieniać układ strony w zależności od potrzeb zdefiniowaliśmy dwa makra `\@setpagelayout` oraz `\@setarticlelayout`. Pierwsze z nich, przelicza nasze prywatne parametry (procentowo wyrażone szerokości marginesów) określone w poprzednim paragrafie, na parametry, których używa L^AT_EX podczas składu strony (szerokość i wysokość tekstu, górny margines, lewy margines strony parzystej i nieparzystej). Drugie makro najpierw ustawia wielkość nagłówka i stopki, a następnie wykonuje `\@setpagelayout`.

Styl strony

W naszej klasie *tga* przewidzieliśmy dwa możliwe style strony. Styl o nazwie `titlepage` odpowiada za wygląd strony tytułowej artykułu, natomiast styl `headings` odpowiada za pozostałe strony. Do składu strony tytułowej wprowadziliśmy dwa pomocnicze makra `\journalinfo` i `\copyrightinfo`. Pierwsze z nich generuje nagłówek dla artykułu zawierający podstawowe dane o naszym czasopiśmie, takie jak: nazwę, numer wolumenu, rok, strony oraz numer ISSN. Drugie makro generuje informację o prawach autorskich, która jest umieszczana w stopce pierwszej strony artykułu.

Strona tytułowa

Podstawowe makro w L^AT_EX-u do utworzenia strony tytułowej to `\maketitle`. Jest ono zdefiniowane w klasie dokumentu i zależy od jego rodzaju. Tak więc, w naszej klasie również definiujemy to makro. Jego zadaniem jest wypisanie tytułu pracy i autorów. Przełącza ono styl bieżącej strony na `titlepage`, co powoduje umieszczenie nagłówka i stopki specyficznych dla naszego czasopisma. Ponadto, w tym miejscu, ustawiane są wartości nagłówków na stronach następnych.

Na stronie tytułowej zwyczajowo umieszcza się abstrakt artykułu, informację o klasyfikacji tematycznej oraz słowa kluczowe. Aby zapewnić prawidłowe formatowanie tych elementów wprowadziliśmy środowisko `abstract` oraz makra: `\msc` i `\keywords`. Makro `\msc` pobiera dwa argumenty – pierwszy z nich to rok klasyfikacji, drugi to numery klasyfikujące rozdzielone przecinkami. Makro `\keywords` pobiera, jako jedyny argument, listę słów kluczowych związanych z treścią artykułu, rozdzielonych przecinkami.

Podział artykułu

Podobnie jak w klasie *article*, w naszej klasie *tga* udostępniamy następujące makra służące do podziału tekstu:

- `\section`,
- `\subsection`,
- `\subsubsection`.

Nie przewidujemy jednak podziału na mniejsze jednostki takie jak *paragraf*, czy *subparagraf*. Wydaje się, że trzy powyższe jednostki są w zupełności wystarczające do składu niedużych objętościowo prac naukowych.

Listy i obiekty ruchome

Jeśli chodzi o listy, obiekty ruchome i parametry ich składu, to ich implementacja została zaczerpnięta z klasy *article*. Jedyna dokonana zmiana polega na uproszczeniu z uwagi na to, że w naszej klasie skład tekstu jest zawsze jednokolumnowy i dwustronny.

Spis literatury

Implementacja środowiska `thebibliography` i jego parametry zostały wzięte z klasy *article*. Zmiana dotyczy rozmiaru czcionki, jaką składany jest spis literatury, na `\footnotesize`. Ponadto został zmniejszony odstęp pionowy `\itemsep` pomiędzy sąsiednimi pozycjami spisu.

Twierdzenia i dowody

W naszej klasie wprowadzamy następujące środowiska do składu twierdzeń i tym podobnych:

<code>thm</code>	–	Theorem,
<code>lem</code>	–	Lemma,
<code>cor</code>	–	Corollary,
<code>rem</code>	–	Remark,
<code>prop</code>	–	Proposition,
<code>fact</code>	–	Fact,
<code>hypo</code>	–	Hypothesis,
<code>conj</code>	–	Conjecture,
<code>crit</code>	–	Criterion,
<code>claim</code>	–	Claim,
<code>axiom</code>	–	Axiom,
<code>assum</code>	–	Assumption,
<code>dfn</code>	–	Definition,
<code>exm</code>	–	Example,
<code>exr</code>	–	Exercise,
<code>prob</code>	–	Problem,
<code>prin</code>	–	Principle,
<code>alg</code>	–	Algorithm,
<code>summ</code>	–	Summary,
<code>conc</code>	–	Conclusion,
<code>note</code>	–	Note,
<code>notat</code>	–	Notation,
<code>hint</code>	–	Hint.

Dzielimy je na trzy grupy, definiując odpowiednio trzy nowe style składu tych obiektów:

normal – pogrubiony nagłówek, treść złożona kursywą – styl składu środowisk: Theorem, Lemma, Corollary, Remark, Proposition, Fact, Hypothesis, Conjecture, Criterion, Claim, Axiom oraz Assumption,

simple – pogrubiony nagłówek, treść złożona prostą czcionką – styl składu środowisk: Definition, Example, Exercise, Problem oraz Principle,

special – pochylony, ale niepogrubiony nagłówek, treść złożona prostą czcionką – styl składu środowisk: Algorithm, Summary, Conclusion, Note, Notation oraz Hint.

Przy definiowaniu tych stylów korzystamy z makra `\newtheoremstyle` dostarczonego w pakiecie *amsthm*. Środowiska składane w stylu *normal* i *simple* są

numerowane. Numerem nadrzędnym jest numer sekcji i wszystkie te środowiska używają wspólnego licznika. Środowiska w stylu *special* nie są numerowane. Tutaj separatorem nagłówka jest dwukropek.

Do składu dowodów służy środowisko `proof`. W zasadzie jest ono zdefiniowane w pakiecie *amsthm*, ale zmieniliśmy krój czcionki jakim składany jest nagłówek (czyli słowo Proof) tego środowiska. Zamiast kroju pochylonego zastosowaliśmy kapitaliki.

Makra pomocnicze

Makra pomocnicze generalnie służą do formatowania pewnych typowych obiektów w tekście, jak na przykład: cytatów, podziękowań, adresów. Jest tam również makro `\setlastpage` używane przy zapisywaniu numeru ostatniej strony artykułu, która pojawia się w nagłówku strony tytułowej.

Inicjalizacja

W tym miejscu wykonywane są procedury inicjujące. Mają one na celu ustawienie domyślnych wartości parametrów składu strony. Za pomocą makra `\AtEndDocument`, na końcu dokumentu zapamiętywany jest numer bieżącej strony. Potrzebny on będzie w nagłówku strony tytułowej.

3.3 Klasa zeszytu

Tak jak pisaliśmy wcześniej, klasa zeszytu, czyli *tgj*, dziedziczy po klasie artykułu *tga*. Tak więc, wszystkie makra zdefiniowane w *tga* są dostępne w *tgj*. W klasie zeszytu definiujemy tylko to co jest tutaj potrzebne, a nie ma tego w klasie artykułu. Niektóre makra zostaną nadpisane.

W dalszej części opisujemy implementację klasy *tgj*, której kod znajduje się w Dodatku B na stronie 36.

Opcje klasy

Ponieważ klasę *tga* ładujemy za pomocą makra `\LoadClassWithOptions`, więc zdefiniowane tam opcje *draft* i *final* będzie można używać w *tgj*. Wprowadzamy tutaj dodatkowo dwie opcje: *cover* i *nocover*. Jak sugerują nazwy, opcje te wzajemnie wykluczają się. Domyślna jest pierwsza z nich, która powoduje że zostaną wygenerowane okładki dla zeszytu.

Dane numeru zeszytu

Nadpisujemy tutaj trzy wewnętrzne makra odpowiadające za rocznik, tom oraz numer zeszytu. W klasie *tga* były one ustawione na stałe, tak aby autor mógł złożyć swój artykuł. Składając natomiast cały zeszyc musimy koniecznie

określić rok, tom i numer zeszytu. Robimy to za pomocą makr: `\tgyear`, `\tgvolume` oraz `\tgnumber`. Mogą one być użyte wyłącznie w preambule.

Deklaracje czcionek

Do stworzenia okładek potrzebujemy kroju pisma o dużym rozmiarze. W klasie *tgj* zdefiniowujemy trzy duże rozmiary czcionek: `\exHuge`, `\ExHuge` oraz `\EXHUGE`.

Układ strony

Zadaniem zdefiniowanego tutaj makra `\@setcoverlayou` jest ustawienie parametrów strony tak, że nie ma nagłówek i stopki. Ten styl strony używany jest podczas składania okładek zeszytu.

Styl strony

Spis treści składany jest na tylnej okładce bez nagłówków i stopki. W tym celu definiujemy nowy styl strony o nazwie `contents`. Jest on stosowany wyłącznie w spisie treści.

Części składowe zeszytu

W zeszycie możemy wyróżnić trzy następujące części: przednią okładkę, część zasadniczą zawierającą artykuły oraz tylną okładkę. Zdefiniowane tutaj makra: `\frontmatter`, `\mainmatter` oraz `\backmatter` służą do odpowiedniego przełączania układu strony oraz stylu numeracji stron w zależności od części zeszytu.

Okładki

Okładki naszego czasopisma zawierają typowe, dla czasopism naukowych, dane. Na przedniej okładce zewnętrznej widnieje nazwa czasopisma, podany jest rok, tom oraz numer. Na dole umieściliśmy wydawcę i numer ISSN. Na stronie wewnętrznej przedniej okładki zamieściliśmy informację o prawach autorskich. Na okładce tylnej umieszczony jest spis treści. Ułatwia to odnalezienie artykułów.

Artykuł

Z punktu widzenia zeszytu artykuł jest podobny do rozdziału książki. Formalnie jednak zdefiniowaliśmy go jako środowisko `article`, nie jako makro proste, którym jest `\chapter` z klasy *book*. Środowisko to pobiera dwa argumenty: tytuł pracy oraz listę autorów. Autorzy na tej liście powinni być rozdzieleni za pomocą makra `\and`. Otwarcie środowiska `article` powoduje utworzenie strony tytułowej artykułu oraz dodanie tego artykułu do spisu

treści poprzez zapisanie danych do pliku `.toc`. Zamknięcie tego środowiska powoduje zapisanie numeru bieżącej strony potrzebnej do podania zakresu stron.

Zdefiniowany jest tutaj również licznik `article`, po to by móc zależnie od niego numerować sekcje, rysunki, tabele itp. Po przejściu do nowego artykułu podane liczniki zostaną skasowane.

Spis treści

Makro `\addtotgcontents` wykorzystywane jest w otwarciu środowiska `article` do wypisania na plik danych do spisu treści. Natomiast wywołanie makra `\tableofcontents` powoduje wypisanie spisu treści zeszytu. Jest tutaj wykorzystane makro z L^AT_EX-a `\@starttoc`, które tworzy spis treści na podstawie pliku `.toc`.

W tej części klasy *tgj* definiujemy również parametry niezbędne do wygenerowania spisu treści.

Makra `\tby` oraz `\@tby` są używane do rozdzielenia tytułu artykułu od listy autorów w pliku `.toc`.

Nadpisujemy tutaj makro `\@sect`, tak aby do spisu treści nie były zapisywane tytuły sekcji. Na spisie pojawiają się wyłącznie tytuły artykułów i ich autorzy.

Makra pomocnicze

Makro `\received` służy do formatowania informacji o tym kiedy artykuł został przyjęty do publikacji. Jest ona umieszczana na końcu artykułu.

Makro `\openoddpag` gwarantuje, że niezależnie od bieżącej pozycji w dokumencie zostanie otwarta nowa, nieparzysta strona.

Inicjalizacja

Na koniec klasy *tgj* wołamy `\AtBeginDocument` oraz `\AtEndDocument`, aby utworzyć okładki zeszytu odpowiednio na jego początku i na końcu, po przetworzeniu wszystkich artykułów. Okładki będą tworzone zgodnie z opcjami jakie były wybrane na początku. Następnie ustawiamy domyślne parametry strony.

3.4 Programy do obróbki zeszytu

W ramach pracy nad czasopismem, w języku Perl, opracowaliśmy skrypt `labels.pl`, który ma za zadanie prefiksować prywatne etykiety użyte w artykule. Jako prefiks używana jest nazwa pliku artykułu. Zakładamy tutaj, że nazwa ta, to identyfikator artykułu nadany mu podczas przyjmowania pracy do publikacji.

Skrypt wykorzystuje parser L^AT_EX-a z opisywanego wcześniej modułu LaTeX::TOM. Pozwala on szybko odszukać wystąpienia poleceń `\label` oraz `\ref` w artykule i umożliwia również modyfikację ich parametrów. Po dodaniu prefiksów, jednym poleceniem `toLaTeX` wypisujemy zmienioną postać pliku L^AT_EX-owego.

Kod skryptu znajduje się w Dodatku C na stronie 42.

Dodatek A

Klasa artykułu tga.cls

```
%%
%% File: tga.cls
%%
%% Description: LaTeX document class to typeset Traditional Geometry article.
%%
%% Create date: Sep 28, 2006
%%
%% Last modified: Sep 28, 2006
%%
%% Copyright (c) 2006 Marzena Kropiewnicka and Mariusz Zynel.
%%
%% This software is FREE. You can use and/or redistribute it for any
%% purpose in either, modified, or unmodified form, under the terms of the
%% GNU General Public License as published by the Free Software Foundation.
%%
%% The above copyright notice and this permission notice shall be included
%% in all copies or substantial portions of this software.
%%
%% THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
%% EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
%% LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.
%%
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{tga}[2006/09/28 v1.0 LaTeX document class (MK/MZ)]

%% Class options

\DeclareOption{draft}{%
  \setlength\overfullrule{5pt}}
\DeclareOption{final}{%
  \setlength\overfullrule{0pt}}

\ExecuteOptions{final}
\ProcessOptions

%% Prerequisites

\RequirePackage{latexsym}[1995/12/01 v2.2e]
\RequirePackage{amsmath}[1997/03/20 v1.2d]
\RequirePackage{amsthm}[1996/10/24 v1.2d]
\RequirePackage{amssymb}[1996/11/03 v2.2b]
\RequirePackage{amsfonts}[1997/09/17 v2.2e]

\input{bk10.clo}
```

%% Journal data

```
\newcommand\tgname{Traditional Geometry}
\newcommand\tgshortname{Trad. Geom.}
\newcommand\issn{ISSN 1234-5678}
\newcommand\copyrightholder{University of Bia{\l}ystok}

\newcommand\tg@year{\number\year}
\newcommand\tg@volume{?}
\newcommand\tg@number{?}
\newcommand\thetgyear{\tg@year}
\newcommand\thetgvolume{\textbf{\tg@volume}}
\newcommand\thetgnumber{\tg@number}
```

%% Font declarations

```
\DeclareOldFontCommand{\rm}{\normalfont\rmfamily}{\mathrm}
\DeclareOldFontCommand{\sf}{\normalfont\sffamily}{\mathsf}
\DeclareOldFontCommand{\tt}{\normalfont\ttfamily}{\mathtt}
\DeclareOldFontCommand{\bf}{\normalfont\bfseries}{\mathbf}
\DeclareOldFontCommand{\it}{\normalfont\itshape}{\mathit}
\DeclareOldFontCommand{\sl}{\normalfont\slshape}{\@nomath\sl}
\DeclareOldFontCommand{\sc}{\normalfont\scshape}{\@nomath\sc}
\DeclareRobustCommand*\cal{\@fontswitch\relax\mathcal}
\DeclareRobustCommand*\mit{\@fontswitch\relax\mathnormal}

\DeclareRobustCommand\tgfamily{%
  \not@math@alphabet\tgfamily\mathrm
  \fontfamily{pag}%
  \selectfont}
\DeclareTextFontCommand{\texttg}{\tgfamily}
```

%% Layout parameters

```
\setlength\paperheight{250mm}
\setlength\paperwidth{176mm}

\def\p@binding{5mm}
\def\p@nonprintablemargin{4mm}
\def\p@horizontalmargin{0.1}
\def\p@verticalmargin{0.1}
\def\p@marginbalance{0.35}
\def\p@abovetitleskip{12ex}
\def\p@aboveauthorskip{3ex}
\def\p@belowauthorskip{6ex}
\def\p@normalheadheight{12pt}
\def\p@normalheadsep{20pt}
\def\p@normalfootskip{30pt}
```

%% Page layout

```
\def\@setarticlelayout{%
  \setlength\headheight{\p@normalheadheight}
  \setlength\headsep{\p@normalheadsep}
  \setlength\footskip{\p@normalfootskip}
  \@setpagelayout
}

\def\@setpagelayout{%
  \setlength\@tempdima{\paperwidth}
  \addtolength\@tempdima{-\p@binding}
```

```

\addtolength \@tempdima      {-\p@horizontalmargin \@tempdima}
\addtolength \@tempdima      {-1in}
\setlength \textwidth        {\@tempdima}
\@settopoint \textwidth

\setlength \@tempdima        {\paperheight}
\addtolength \@tempdima      {-\p@verticalmargin \@tempdima}
\addtolength \@tempdima      {-\headheight}
\addtolength \@tempdima      {-\headsep}
\addtolength \@tempdima      {-\footskip}
\divide \@tempdima \baselineskip
\@tempcnta=\@tempdima
\setlength \textheight       {\@tempcnta \baselineskip}
\addtolength \textheight     {\topskip}
\@settopoint \textheight

\setlength \@tempdima        {\paperwidth}
\addtolength \@tempdima      {-\p@binding}
\addtolength \@tempdima      {-\textwidth}
\setlength \oddsidemargin    {\p@marginbalance \@tempdima}
\addtolength \oddsidemargin  {\p@binding}
\addtolength \oddsidemargin  {-1in}
\@settopoint \oddsidemargin

\setlength \evensidemargin   {\paperwidth}
\addtolength \evensidemargin {-\textwidth}
\addtolength \evensidemargin {-\oddsidemargin}
\addtolength \evensidemargin {-2in}
\@settopoint \evensidemargin

\setlength \marginparwidth   {\evensidemargin}
\addtolength \marginparwidth {-\p@nonprintablemargin}
\addtolength \marginparwidth {-\marginparsep}
\@settopoint \marginparwidth

\setlength \topmargin         {\paperheight}
\addtolength \topmargin      {-\headheight}
\addtolength \topmargin      {-\headsep}
\addtolength \topmargin      {-\textheight}
\addtolength \topmargin      {-\footskip}
\addtolength \topmargin      {-0.5\topmargin}
\addtolength \topmargin      {-1in}
\@settopoint \topmargin

\@colht \textheight
\@colroom \textheight
\vsizel \textheight
\hsizel \textwidth
\linewidth \textwidth
}

```

%% Page style

```

\newcommand \journalinfo {%
  \normalfont
  \setbox \@tempboxa = \hbox { \tgfamily \small \bfseries \tgname } %
  \hbox { \scriptsize
    \vbox to \headheight {%
      \vfill
      \hbox { \tgshortname \space \thetgvolume \space ( \thetgyear ) %
        \thepage \space - \space \lastpage } \vskip 1\p@
      \hbox { \issn }
      \vss }
    }
  \hss }
\hfil

```

```

\hbox to\wd\@tempboxa{%
\ vbox to\headheight{%
\ vfill
\usebox\@tempboxa\ vskip2\p@
\hrule height 0.4pt depth 0pt width \wd\@tempboxa
\ vss}
\hss}
}

\newcommand\copyrightinfo{%
\hbox{\normalfont\tiny\copyright\space\copyrightholder,\space\thetgyear}
}

\def\ps@headings{%
\def\@evenhead{\reset@font\small%
\rlap{\thepage}\hfil\leftmark\hfil\llap{\tgshortname}}
\let\@evenfoot\@empty
\def\@oddhead{\reset@font\small%
\rlap{Vol.\space\thetgvolume\space(\thetgyear)}%
\hfil\rightmark\hfil\llap{\thepage}}
\let\@oddfoot\@empty
\let\@mkboth\markboth
\let\@mkright\markright
}%

\def\ps@titlepage{%
\let\@mkboth\@gobbletwo
\def\@oddhead{\journalinfo}
\def\@oddfoot{\rlap{\copyrightinfo}\hfil\thepage\hfil}
}

%% Title page

\newcommand\maketitle{
\par
\begingroup
\renewcommand\thefootnote{\@fnsymbol\c@footnote}%
\def\@makefnmark{\rlap{\@textsuperscript{\normalfont\@thefnmark}}\space}%
\long\def\@makefntext##1{\parindent 1em\noindent
\hb@xt@1.8em{%
\hss\@textsuperscript{\normalfont\@thefnmark}}##1}%
\newpage
\global\@topnum\z@ % Prevents figures from going at top of page.
\thispagestyle{titlepage}
\@maketitle
\@thanks
\let\thanks\@gobble
\let\footnote\thanks
\@mkboth{\@author}{\@title}
\endgroup
\setcounter{footnote}{0}%
\global\let\@thanks\@empty
\global\let\@author\@empty
\global\let\@title\@empty
}

\def\@maketitle{%
\null
\vskip\p@abovetitleskip
\begingroup
\parindent\z@
\let\footnoterule\relax
\let\footnote\thanks
{\large\bfseries\@title}
\par

```



```

        \vskip\p@aboveauthorskip
        {\scshape\@author}
    \endgroup
    \par
    \vskip\p@belowauthorskip
}

\renewcommand\and{%
    {\normalfont\andname}\space
}

\newenvironment{abstract}{%
    \par
    \addvspace\medskipamount
    \noindent
    \small
    {\itshape Abstract}.\space
    \ignorespaces
}{%
    \vskip\medskipamount
}

\newcommand\msc[2]{%
    \par
    \addvspace\medskipamount
    \noindent
    {\small {\itshape Mathematics Subject Classification} (#1): #2.}
}

\newcommand\keywords[1]{%
    \par
    \noindent
    {\small {\itshape Key words}: #1.}
    \vskip\medskipamount
}

%% Sectioning

\setcounter{secnumdepth}{3}
\newcounter{section}
\newcounter{subsection}[section]
\newcounter{subsubsection}[subsection]
\renewcommand\thesection    {\@arabic\c@section}
\renewcommand\thesubsection {\thesection.\@arabic\c@subsection}
\renewcommand\thesubsubsection{\thesubsection.\@arabic\c@subsubsection}

\newcommand\section{\@startsection{section}{1}{\z@}%
    {-3.5ex \@plus -1ex \@minus -.2ex}%
    {2.3ex \@plus .2ex}%
    {\normalfont\large\bfseries}}

\newcommand\subsection{\@startsection{subsection}{2}{\z@}%
    {-3.25ex \@plus -1ex \@minus -.2ex}%
    {1.5ex \@plus .2ex}%
    {\normalfont\normalsize\bfseries}}

\newcommand\subsubsection{\@startsection{subsubsection}{3}{\z@}%
    {-3.25ex \@plus -1ex \@minus -.2ex}%
    {1.5ex \@plus .2ex}%
    {\normalfont\normalsize\bfseries}}

%% Lists

\setlength\leftmargini {2.5em}

```

```

\leftmargin \leftmargini
\setlength\leftmarginii {2.2em}
\setlength\leftmarginiii {1.87em}
\setlength\leftmarginiv {1.7em}
\setlength\leftmarginv {1em}
\setlength\leftmarginvi {1em}
\setlength \labelsep {1.5em}
\setlength \labelwidth{\leftmargini}
\addtolength\labelwidth{-\labelsep}
\@beginparpenalty -\@lowpenalty
\@endparpenalty -\@lowpenalty
\@itempenalty -\@lowpenalty
\renewcommand\theenumi{\@arabic\c@enumi}
\renewcommand\theenumii{\@alph\c@enumii}
\renewcommand\theenumiii{\@roman\c@enumiii}
\renewcommand\theenumiv{\@Alph\c@enumiv}
\newcommand\labelenumi{\theenumi.}
\newcommand\labelenumii{\theenumii}
\newcommand\labelenumiii{\theenumiii.}
\newcommand\labelenumiv{\theenumiv.}
\renewcommand\p@enumii{\theenumi}
\renewcommand\p@enumiii{\theenumi(\theenumii)}
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
\newcommand\labelitemi{\textbullet}
\newcommand\labelitemii{\normalfont\bfseries \textendash}
\newcommand\labelitemiii{\textasteriskcentered}
\newcommand\labelitemiv{\textperiodcentered}

\newenvironment{description}{%
  \list{}{\labelwidth \z@
    \itemindent -\leftmargin
    \let\makelabel \descriptionlabel}%
}{%
  \endlist
}

\newcommand*\descriptionlabel[1]{%
  \hspace\labelsep
  \normalfont\bfseries #1%
}

%% Floats

\setlength\lineskip{1\p@}
\setlength\normallineskip{1\p@}
\renewcommand\baselinestretch{}
\setlength\parskip{0\p@ \@plus \p@}
\@lowpenalty 51
\@medpenalty 151
\@highpenalty 301
\setcounter{topnumber}{2}
\renewcommand\topfraction{.7}
\setcounter{bottomnumber}{1}
\renewcommand\bottomfraction{.3}
\setcounter{totalnumber}{3}
\renewcommand\textfraction{.2}
\renewcommand\floatpagefraction{.5}
\setcounter{dbltopnumber}{2}
\renewcommand\dbltopfraction{.7}
\renewcommand\dblfloatpagefraction{.5}

\setlength\arraycolsep{5\p@}
\setlength\tabcolsep{6\p@}
\setlength\arrayrulewidth{.4\p@}
\setlength\doublerulesep{2\p@}

```

```

\setlength\tabbingsep{\labelsep}
\skip\@mpfootins = \skip\footins
\setlength\fbboxsep{3\p@}
\setlength\fbboxrule{.4\p@}
\renewcommand\theequation{\@arabic\c@equation}
\newcounter{figure}
\renewcommand\thefigure{\@arabic\c@figure}
\def\fps@figure{tbp}
\def\ftype@figure{1}
\def\ext@figure{lof}
\def\fnm@figure{\figurename~\thefigure}
\newenvironment{figure}
  {\@float{figure}}
  {\end@float}
\newenvironment{figure*}
  {\@dblfloat{figure}}
  {\end@dblfloat}
\newcounter{table}
\renewcommand\thetable{\@arabic\c@table}
\def\fps@table{tbp}
\def\ftype@table{2}
\def\ext@table{lot}
\def\fnm@table{\tablename~\thetable}
\newenvironment{table}
  {\@float{table}}
  {\end@float}
\newenvironment{table*}
  {\@dblfloat{table}}
  {\end@dblfloat}
\newlength\abovecaptionskip
\newlength\belowcaptionskip
\setlength\abovecaptionskip{10\p@}
\setlength\belowcaptionskip{0\p@}
\long\def\@makecaption#1#2{%
  \vskip\abovecaptionskip
  \sbox\@tempboxa{#1: #2}%
  \ifdim \wd\@tempboxa >\hsizel
    #1: #2\par
  \else
    \global \@minipagefalse
    \hb@xt@\hsizel{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}

%% References

\newdimen\bibindent
\setlength\bibindent{1.5em}
\newenvironment{thebibliography}[1]
  {\section*{\bibname}%
  \list{\@biblabel{\@arabic\c@enumiv}}%
  {\settowidth\labelwidth{\@biblabel{#1}}%
  \leftmargin\labelwidth
  \advance\leftmargin\labelsep
  \itemsep\z@
  \footnotesize
  \@openbib@code
  \usecounter{enumiv}%
  \let\p@enumiv\@empty
  \renewcommand\theenumiv{\@arabic\c@enumiv}}%
  \sloppy
  \clubpenalty4000
  \@clubpenalty\clubpenalty
  \widowpenalty4000%
  \sfcode'\.\@m}

```

```

{\def\@noitemerr
  {\@latex@warning{Empty ‘thebibliography’ environment}}%
  \@endparpenalty\@M\endlist}
\newcommand\newblock{\hskip .11em\@plus.33em\@minus.07em}
\let\@openbib@code\@empty

```

%% Theorem environments

```

\newtheoremstyle{normal}
  {}%                space above
  {}%                space below
  {\itshape}%       body font
  {}%                indent amount
  {\bfseries}%      head font
  {.%}%             punctuation
  {5\p@ plus\p@ minus\p@}% space after head
  {}%                head spec

```

```

\newtheoremstyle{simple}
  {}%                space above
  {}%                space below
  {}%                body font
  {}%                indent amount
  {\bfseries}%      head font
  {.%}%             punctuation
  {5\p@ plus\p@ minus\p@}% space after head
  {}%                head spec

```

```

\newtheoremstyle{special}
  {}%                space above
  {}%                space below
  {}%                body font
  {}%                indent amount
  {\itshape}%       head font
  {:}%              punctuation
  {5\p@ plus\p@ minus\p@}% space after head
  {}%                head spec

```

`\theoremstyle{normal}`

```

\newtheorem{thm}{Theorem}[section]
\newtheorem{lem}[thm]{Lemma}
\newtheorem{cor}[thm]{Corollary}
\newtheorem{rem}[thm]{Remark}
\newtheorem{prop}[thm]{Proposition}
\newtheorem{fact}[thm]{Fact}
\newtheorem{hypo}[thm]{Hypothesis}
\newtheorem{conj}[thm]{Conjecture}
\newtheorem{crit}[thm]{Criterion}
\newtheorem{claim}[thm]{Claim}
\newtheorem{axiom}[thm]{Axiom}
\newtheorem{assum}[thm]{Assumption}

```

`\theoremstyle{simple}`

```

\newtheorem{dfn}[thm]{Definition}
\newtheorem{exm}[thm]{Example}
\newtheorem{exr}[thm]{Exercise}
\newtheorem{prob}[thm]{Problem}
\newtheorem{prin}[thm]{Principle}

```

`\theoremstyle{special}`

```

\newtheorem*{alg}{Algorithm}
\newtheorem*{summ}{Summary}

```

```

\newtheorem*{conc}{Conclusion}
\newtheorem*{note}{Note}
\newtheorem*{notat}{Notation}
\newtheorem*{hint}{Hint}

\renewenvironment{proof}[1][Proof]{%
  \par
  \normalfont
  \topsep6\p@\@plus6\p@ \trivlist
  \item[\hskip\labelsep\scshape #1\@addpunct{.}]{\ignorespaces}
}{%
  \qed\endtrivlist
}

%% Utilities

\newenvironment{quotation}{%
  \list{}{\listparindent 1.5em%
    \itemindent \listparindent
    \rightmargin \leftmargin
    \parsep \z@ \@plus\p@}%
  \item\relax%
}{%
  \endlist
}

\newenvironment{quote}{%
  \list{}{\rightmargin \leftmargin}%
  \item\relax%
}{%
  \endlist
}

\newenvironment{acknowledgments}{%
  \par
  \addvspace\medskipamount
  \noindent
  {\itshape Acknowledgements}.\space
  \ignorespaces
}{%
  \vskip\medskipamount
}

\newenvironment{address}{%
  \noindent
  \footnotesize
  \begin{flushleft}
}{%
  \end{flushleft}
}

\newcommand\email[1]{%
  e-mail: \texttt{#1}
}

\renewcommand\footnoterule{%
  \kern-3\p@
  \hrule\@width.4\columnwidth
  \kern2.6\p@}

\newcommand\@makefntext[1]{%
  \parindent 1em%
  \noindent
  \hb@xt@1.8em{\hss\@makefnmark}#1}

```

```
\def\lastpage{??}

\newcommand\setlastpage[1]{%
  \gdef\lastpage{#1}
}

%% Class error routine

\newcommand{\tga@classerr}{\ClassError{TGA}}

%% Constant names

\newcommand\bibname{References}
\newcommand\figurename{Figure}
\newcommand\tablename{Table}
\newcommand\volumentname{Volume}
\newcommand\numbername{Number}
\newcommand\etalname{et al.}
\newcommand\byname{By}
\newcommand\andname{and}

%% Initialization

\AtEndDocument{%
  \protected@write\@auxout{}%
    {\string\setlastpage{\thepage}}
}

\@twosidetrue
\@mparswitchtrue
\@setarticlelayout
\pagestyle{headings}
\pagenumbering{arabic}
\raggedbottom
\onecolumn
\endinput
%%
%% End of file 'tga.cls'.
```

Dodatek B

Klasa zaszytu tgja.cls

```
%%
%% File: tgj.cls
%%
%% Description: LaTeX document class to typeset Traditional Geometry journal.
%%
%% Create date: Sep 28, 2006
%%
%% Last modified: Sep 28, 2006
%%
%% Copyright (c) 2006 Marzena Kropiewnicka and Mariusz Zynel.
%%
%% This software is FREE. You can use and/or redistribute it for any
%% purpose in either, modified, or unmodified form, under the terms of the
%% GNU General Public License as published by the Free Software Foundation.
%%
%% The above copyright notice and this permission notice shall be included
%% in all copies or substantial portions of this software.
%%
%% THIS SOFTWARE IS PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND,
%% EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDER BE
%% LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.
%%
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{tgj}[2006/09/28 v1.0 LaTeX document class (MK/MZ)]

%% Class options

\DeclareOption{cover}{%
  \def\at@begindocument{\frontmatter\frontcover\mainmatter}
  \def\at@enddocument{\backmatter\backcover}}
\DeclareOption{nocover}{%
  \def\at@begindocument{\mainmatter}
  \let\at@enddocument\relax}

\ExecuteOptions{final,cover}
\ProcessOptions
\LoadClassWithOptions{tga}

%% Journal data

\newcommand\tg@year{\undef@err{\tgyear}}
\newcommand\tg@volume{\undef@err{\tgvolume}}
\newcommand\tg@number{\undef@err{\tgnumber}}
\newcommand\tgyear[1]{\gdef\tg@year{#1}}
```

```

\newcommand\tgvolume[1]{\gdef\tg@volume{#1}}
\newcommand\tgnumber[1]{\gdef\tg@number{#1}}
\@onlypreamble\tgyear
\@onlypreamble\tgvolume
\@onlypreamble\tgnumber

%% Font declarations

\newcommand\exHuge{\@setfontsize\exHuge{44}{50}}
\newcommand\ExHuge{\@setfontsize\ExHuge{64}{72}}
\newcommand\EXHUGE{\@setfontsize\EXHUGE{72}{96}}

%% Page layout

\def\@setcoverlayout{%
  \setlength\headheight{\z@}
  \setlength\headsep{\z@}
  \setlength\footskip{\z@}
  \@setpagelayout}

%% Page style

\def\ps@contents{%
  \let\@mkboth\@gobbletwo
  \let\@oddhead\@empty
  \let\@oddfoot\@empty
  \let\@evenhead\@empty
  \def\@evenfoot{%
    \ifnum\c@article>\toclength
      \small\itshape\hfill\continuedname
    \else
      \@empty\relax
    \fi}}

%% Booklet parts

\newif\if@mainmatter \@mainmattertrue

\newcommand\frontmatter{%
  \@setcoverlayout
  \cleardoublepage
  \@mainmatterfalse
  \pagenumbering{roman}
  \setcounter{page}\@ne
}

\newcommand\mainmatter{%
  \@setarticlelayout
  \cleardoublepage
  \@mainmattertrue
  \pagenumbering{arabic}
  \setcounter{page}\@ne
}

\newcommand\backmatter{%
  \cleardoublepage
  \@setcoverlayout
  \@mainmatterfalse
}

%% Covers

```

```

\newcommand\outsidefrontcover{%
  \thispagestyle{empty}
  \parindent\z@
  \normalfont
  \hfill{\tgfamily\EXHUGE\tg@number}\par
  \vskip 24\p@
  \hfill{\huge\bfseries Volume \tg@volume, \tg@year}\par
  \vskip 48\p@
  \begin{center}
    {\tgfamily\ExHuge Traditional\[\[15 pt] Geometry}
    \vfill
  \end{center}
  \bfseries \copyrightholder\par
  \bigskip
  \issn
  \end{center}
}

\newcommand\insidefrontcover{%
  \thispagestyle{empty}
  \parindent\z@
  \normalfont
  \null
  \vfill
  \begin{flushleft}
    \baselineskip 15\p@
    \texttt{\tgname}\par
    \medskip
    Copyright {\copyright \tg@year} by \copyrightholder.\
    All rights reserved.\
    \issn.
  \end{flushleft}
}

\newcommand\frontcover{%
  \outsidefrontcover
  \newpage
  \insidefrontcover
}

\newcommand\insidebackcover{%
}

\newcommand\outsidebackcover{%
  \tableofcontents
}

\newcommand\backcover{%
  \insidebackcover
  \newpage
  \outsidebackcover
}

%% Article

\newcounter{article}
\renewcommand\thearticle{\@arabic\c@article}

\newenvironment{article}[2]{%
  \title{#1}
  \author{#2}
  \cleardoublepage
  \maketitle
  \refstepcounter{article}%
  \begingroup

```

```

        \def\@protect{\protect}
        \let\thanks\@gobble
        \let\footnote\thanks
        \typeout{Article: #1}%
        \addtotgcontents{#1\protect{\tby #2\protect}}%
    \endgroup
    \@afterheading
}{%
    \setlastpage{\thepage}
}

%% Counters

\@addtoreset{equation}{article}
\@addtoreset{figure}{article}
\@addtoreset{table}{article}

%% Table of contents

\newcount\toclength \toclength 13

\newcommand\@pnumwidth{1.55em}
\newcommand\@tocmarg{2.55em}
\newcommand\@dotsep{4.5}
\setcounter{tocdepth}{2}
\newcommand\tableofcontents{%
    \parindent\z@
    \pagestyle{contents}
    \vspace*{40\p@}
    \@starttoc{to1}%
    \clearpage
    {\LARGE\bfseries\contentsname}\hfill%
    {\small{\itshape\tgshortname} \thetgvolume (\thetgyear)}\par
    \vspace*{40\p@}
    \@starttoc{toc}
}

\newcommand*\l@article[2]{%
    \ifnum \c@tocdepth >\m@ne
        \addpenalty{-\@highpenalty}%
        \vskip 1.0em \@plus\p@
        \setlength\@tempdima{1.5em}%
        \begingroup
            \parindent \z@ \rightskip \@pnumwidth
            \parfillskip -\@pnumwidth
            \leavevmode \bfseries
            \advance\leftskip\@tempdima
            \hskip -\leftskip
            #1\nobreak\mdseries
            \leaders\hbox{\$ \m@th
                \mkern \@dotsep mu\hbox{.}\mkern \@dotsep
                mu$}\hfill
            \nobreak\bfseries
            \hb@xt@\@pnumwidth{\hss #2}\par
            \penalty\@highpenalty
        \endgroup
    \fi}

\newcommand\addtotgcontents[1]{%
    \ifnum \c@article >\toclength
        \addcontentsline{toi}{article}{#1}
    \else
        \addcontentsline{toc}{article}{#1}
    \fi
}

```

```

\def\@tby{\mdseries\scshape }
\def\tby{\@protect\@tby}

\def\@sect#1#2#3#4#5#6[#7]#8{%
  \ifnum #2>\c@secnumdepth
    \let\@svsec\@empty
  \else
    \refstepcounter{#1}%
    \protected@edef\@svsec{\@secntformat{#1}\relax}%
  \fi
  \@tempskipa #5\relax
  \ifdim \@tempskipa > \z@
    \begingroup
      #6{%
        \@hangfrom{\hskip #3\relax\@svsec}%
        \interlinepenalty \@M #8\@@par}%
    \endgroup
    \csname #1mark\endcsname{#7}%
  \else
    \def\@svsechd{%
      #6{\hskip #3\relax
        \@svsec #8}%
      \csname #1mark\endcsname{#7}}%
  \fi
  \@xsect{#5}}

%% Utilities

\newcommand\received[1]{%
  \@beginparpenalty \@M
  \@endparpenalty \@M
  \advspace{8ex}%
  \parindent\z@
  \normalfont
  \begin{center}
    \vbox{\slshape \receivedname \ #1\par
      \advspace{4\p@}
      \rule{4em}{.4\p@}}
  \end{center}
  \@beginparpenalty -\@lowpenalty
  \@endparpenalty -\@lowpenalty}

\def\openoddpage{\clearpage%
  \if@twoside
    \ifodd\c@page
  \else
    \thispagestyle{empty}
    \hbox{}\newpage
  \if@twocolumn
    \hbox{}\newpage
  \fi
  \fi}
\let\cleardoublepage\openoddpage

%% Class error routine

\newcommand{\tgj@classerr}{\ClassError{TGJ}}
\def\undef@err#1{\tgj@classerr{%
  Undefined \protect#1%
}}{%
  Terminate and complete description of the TG issue.%
}}%

```

%% Constant names

```
\newcommand\contentsname{Contents}  
\newcommand\receivedname{Received}  
\newcommand\continuedname{Continued on inside back cover}
```

%% Initialization

```
\AtBeginDocument{\at@begindocument}  
\AtEndDocument{\at@enddocument}
```

```
\@twosidetrue  
\@mparswitchtrue  
\@setarticlayout  
\pagestyle{headings}  
\pagenumbering{arabic}  
\raggedbottom  
\onecolumn
```

```
\endinput
```

```
%%  
%% End of file 'tgj.cls'.
```

Dodatek C

Program labels.cls

```
#!/opt/cfw/bin/perl

use LaTeX::TOM;

my $latexfile = $ARGV[0];

(my $prefix = $latexfile) =~ s/\.tex//;

my $parser = new Parser;
my $article = $parser->parseFile($latexfile);

my $labels = $article->getCommandNodesByName("label");

foreach my $item (@$labels) {
    my $children = $item->getChildTree;
    foreach my $child (@$children) {
        $child->setNodeText($prefix . "_" . $child->getNodeText);
    }
}

my $refs = $article->getCommandNodesByName("ref");

foreach my $item (@$refs) {
    my $children = $item->getChildTree;
    foreach my $child (@$children) {
        $child->setNodeText($prefix . "_" . $child->getNodeText);
    }
}

print $article->toLaTeX;
```

Bibliografia

- [1] Lamport L., *LaTeX system przygotowywania dokumentów*, Wydawnictwo Naukowo-Techniczne, Warszawa, 2004.
- [2] Partl H., Hyna I. i Schlegl E., *Nie za krótkie wprowadzenie do systemu LaTeX2E*, wersja 3.2, wrzesień 1998.
- [3] Strona domowa projektu *LaTeX::TOM*:
http://br.endernet.org/~akrowne/elaine/latex_tom/
- [4] Gajda W., HTML kontra LaTeX:
www.gajdaw.pl/latex/html-kontra-latex.html
- [5] Wikipedia, Regular expression:
http://en.wikipedia.org/wiki/Regular_expression
- [6] Wikipedia, Wyrażenia regularne w Perlu:
http://wiki.alrauna.org/index.php/Artykuły/Wyrażenia_reguluarne_w_Perlu