

## SPIS TREŚCI

1. Krótka instrukcja obsługi .....	2
1.1. Jak użyć szablonu .....	2
1.2. Pisanie w $\text{\LaTeX}$ 'u .....	2
1.3. Organizacja projektu.....	2
1.4. Strona tytułowa i oświadczenie .....	3
1.5. Język polski i kodowanie.....	4
1.5.1. Kilka słów wyjaśnienia .....	4
1.6. Matematyka .....	4
1.6.1. Środowiska twierdzeń.....	4
1.7. Wyliczenia.....	5
1.8. Obrazki.....	6
1.9. Dodawanie listingu kodu .....	8
1.10 Dodawanie bibliografii .....	9
1.11 Uwagi o typografii .....	10
2. Uwagi od strony technicznej.....	11
2.1. Opcje .....	11
2.2. Wymagane pakiety .....	11
2.3. Fonty .....	11
Wykaz literatury .....	12
Wykaz rysunków .....	12
Wykaz tabel .....	13

## 1. KRÓTKA INSTRUKCJA OBSŁUGI

*Do what you think is interesting, do something that you think is fun and worthwhile, because otherwise you won't do it well anyway.* —Brian W. Kernighan

Pakiet SzablonPG powstał, aby ułatwić pisanie prac dyplomowych na Politechnice Gdańskiej. Ułatwienie to ma polegać na podaniu szablonu  $\text{\LaTeX}$ owego pracy dyplomowej, który będzie maksymalnie zgodny z wytycznymi dla autorów podanych w Zarządzeniu Rektora 22/2018 z 20 czerwca 2018 roku. W ten sposób zamiast „walczyć” z  $\text{\LaTeX}$ em lub starać się złożyć pracę w MS Word można skupić się na treści. Korzystanie z  $\text{\LaTeX}$ a szczególnie w przypadku prac matematycznych wydaje się być wskazane z uwagi na łatwość składania i jakość złożonych formuł matematycznych.

### 1.1. Jak użyć szablonu

Szablon dostarcza klasy SzablonPG, którą należy użyć zamiast standardowych klas np.: `article`, `report`. Plik `szablonPG.cls` **musi być** umieszczony w tym samym katalogu, co główny plik pracy. Na początku głównego pliku pracy piszemy

```
\documentclass{szablonPG} % zamiast standardowych klas np.: article
```

Plik `SzablonPracyPG_main.tex` pokazuje jak może wyglądać praca złożona z użyciem tej klasy.

### 1.2. Pisanie w $\text{\LaTeX}$ u

Aby wygodnie pisać w  $\text{\LaTeX}$ u potrzebujemy:

1. dystrybucji  $\text{\TeX}$ a np.: MikTeX lub TeXLive (Linux)
2. edytora np.: LED, WinEdit, Kile (Linux)

Podstawowym źródłem wiedzy o programowaniu jest dostarczana z każdą dystrybucją dokumentacja:

1. A Gentle Introduction to TEX,
2. The Not So Short Introduction to LATEX

Internet jest pełen wprowadzeń, tutoriali i przykładów. Nieocenionym źródłem wiedzy są strony:

```
http://google.com/  
http://faq.gust.org.pl/  
http://tex.stackexchange.com/
```

### 1.3. Organizacja projektu

Pisanie w jednym pliku dużego projektu, powoduje, że w pewnym momencie ciężko odszukać wśród setek linii kodu tą właściwą. System  $\text{\LaTeX}$ umożliwia podzielenie kodu źródłowego na osobne pliki, służą do tego polecenia `\input` oraz `\include`. Opiszemy sposób korzystania z tego drugiego polecenia.

Na początek dzielimy naszą pracę na logiczne fragmenty. W przypadku pracy dyplomowej mogą to być np.: rozdziały. Następnie tworzymy osobne pliki  $\text{\TeX}$ 'owe dla każdego rozdziału i jeden plik główny.

W pliku głównym umieszczamy wszystkie deklaracje, dołączamy pakiety, których chcemy użyć i sekwencję `\begin{document} ... \end{document}`. W tej sekwencji za pomocą polecenia `\include` dołączamy pliki rozdziałów. W plikach rozdziałów piszemy **tylko kod rozdziału**.

Założmy, że nasza praca składa się ze *Wstępu*, *Rozdziału 1* i *Rozdziału 2*. Tworzymy wtedy pliki (nazwy plików przykładowe): `praca_glowny.tex`, `praca_wstep.tex`, `praca_rozdzial1.tex`, `praca_rozdzial2.tex`. UWAGA!  $\text{\LaTeX}$  nie lubi spacji w nazwach plików. Przykładowo, pliki mogłyby wyglądać tak:

<code>% praca_glowny.tex</code>	<code>% praca_wstep.tex</code>
<code>\documentclass{SzablonPracyPG}</code>	<code>\chapter*{Wstęp}</code>
<code>%inne pakiety, definicje itp.</code>	
<code>\begin{document}</code>	<code>To jest wstęp.</code>
<code>\tableofcontents</code>	
<code>\include{praca_wstep}</code>	<code>% praca_rozdzial1.tex</code>
<code>\include{praca_rozdzial1}</code>	<code>\chapter{Tytuł pierwszego rozdziału}</code>
<code>\include{praca_rozdzial2}</code>	
<code>% bibilografia etc.</code>	<code>Treść pierwszego rozdziału.</code>
<code>\end{document}</code>	<code>% praca_rozdzial2.tex</code>
<code>%EOF</code>	<code>\chapter{Tytuł drugie rozdziału}</code>
	<code>Treść drugiego rozdziału.</code>

Pliki nie muszą być umieszczone w tym samym katalogu co plik główny, trzeba wtedy poprzedzić nazwę pliku ścieżką do katalogu w którym znajduje się plik. Przykładowo można w tym celu użyć następującego polecenia: `\input{rozdzialy/praca_rozdzial1.tex}`. Wszystkie edytory  $\text{\TeX}$ 'owe wspierają tworzenie projektów, czyli prace w ten sposób, pomagając w ten sposób zapamiętać nad strukturą pracy. Dla długich projektów przydatne jest także polecenie `\includeonly`.

#### 1.4. Strona tytułowa i oświadczenie

Najprościej - umieszczamy w katalogu **meta** pliki oświadczenia oraz strony tytułowej wygenerowane z serwisu MojaPG, zapisane po uzupełnieniu danych w formacie pdf. Następnie w głównym pliku pracy dyplomowej dodajemy pliki do całego dokumentu przy użyciu poleceń:

```
%-----
% Dodanie strony tytułowej wygenerowanej z MojaPG oraz
% oświadczenia
\includepdf{meta/strona_tytulowa.pdf}
\includepdf{meta/oswiadczenie.pdf}
%-----
```

Polecenia te znajdują się już w przykładowym pliku głównym pracy `SzablonPracyPG_main.tex`.

## 1.5. Język polski i kodowanie

W przypadku, gdy:

1. plik SzablonPracyPG\_main.tex nie chce się kompilować,
2. kod źródłowy tego przykładu zawiera „krzaczki”,
3. widoczny po skompilowaniu tekst: nie zawiera polskich liter,
4. zamiast polskich liter wyświetla „krzaczki”,

to jedną z przyczyn może być niewłaściwe kodowanie polskich znaków. Domyślnie przyjmujemy kodowanie windowsowe: cp1250. Zmiana jest możliwa poprzez ręczną edycję linii

```
\RequirePackage[cp1250]{inputenc}
```

Możliwe wartości to:

1. `\RequirePackage[utf8]{inputenc}` (dla kodowania utf8)
2. `\RequirePackage[latin2]{inputenc}` (dla kodowania iso8859-2)

Inną możliwością jest zmiana kodowania pliku w używanym edytorze.

### 1.5.1. Kilka słów wyjaśnienia

Jedną z wielkich bolączek typografii komputerowej jest/był brak jednolitego formatu kodowania polskich znaków (lub ogólniej znaków innych niż występujących w angielskich tekstach). We wczesnych latach pojawiło się **bardzo dużo** niezgodnych ze sobą systemów. Pewną standaryzację wprowadziły DOS i Windows (strony kodowe cpXXXX) i upowszechnienie WWW (warianty isoXXXX-X). Obecnie coraz powszechniejsze jest kodowanie w standardzie utfX.

## 1.6. Matematyka

### 1.6.1. Środowiska twierdzeń

W klasie zdefiniowane są następujące środowiska matematyczne;

- Definicja

**Definicja 1.1.** Liczbę naturalną, nazywamy liczbą pierwszą, jeżeli jej jedynymi dzielnikami jest 1 i ona sama.

```
\begin{definicja}
Liczbę naturalną, nazywamy liczbą pierwszą, jeżeli jej jedynymi dzielnikami
jest  $1$  i ona sama.
\end{definicja}
```

- Lemat

**Lemat 1.2.** Niech  $p$  będzie liczbą pierwszą. Jeżeli  $p|ab$ , to  $p|a$  lub  $p|b$ .

```
\begin{lemat}
Niech  $p$  będzie liczbą pierwszą. Jeżeli  $p|ab$ , to  $p|a$  lub  $p|b$ .
\end{lemat}
```

- Twierdzenie

**Twierdzenie 1.3.** Liczb pierwszych jest nieskończenie wiele.

```
\begin{twierdzenie}
Liczba pierwszych jest nieskończenie wiele.
\end{twierdzenie}
```

- Wniosek

**Wniosek 1.4.** Ciąg liczb pierwszych jest nieograniczony.

```
\begin{wniosek}
Ciąg liczb pierwszych jest nieograniczony.
\end{wniosek}
```

- Przykład

**Przykład 1.5.**

Prowadzona numeracja środowisk jest ciągła, aby ułatwić odnajdywanie numerów w tekście. Wszystkie środowiska złożone są w ten sam sposób, modyfikacja jest możliwa poprzez utworzenie własnego stylu i zadeklarowanie jego użycia przed zdefiniowaniem środowiska (patrz. dokumentacja pakietów `ams*` i plik klasy).

Do twierdzeń, definicji itd. możemy odwoływać się korzystając z konstrukcji `\label – \ref`, np.; Definicja 1.1 jest bardzo ważna. Twierdzenie 1.3 było znane już Euklidesowi.

Definicja `\ref{def:pierwsza}` jest bardzo ważna.

Twierdzenie `\ref{tw:ilepierwszych}` było znane już Euklidesowi.

## 1.7. Wyliczenia

Wyliczenia numerowane tworzymy za pomocą następującego polecenia:

```
\begin{enumerate}
\item Jeżeli  $2|n$ , to  $n$  jest liczbą parzystą.
\item Liczba parzystych jest nieskończenie wiele.
\item Istnieje dokładnie jedna parzysta liczba pierwsza.
\end{enumerate}
```

1. Jeżeli  $2|n$ , to  $n$  jest liczbą parzystą.
2. Liczba parzystych jest nieskończenie wiele.
3. Istnieje dokładnie jedna parzysta liczba pierwsza.

Jeżeli chcemy uniknąć wcięcia akapitowego bezpośrednio po wyliczeniu należy zastosować polecenie `\noindent`. Do list numerowanych można dodać dodatkowy parametr `label=`, np.:

```
\begin{definicja}
\emph{Grupą} nazywamy parę  $(G, \circ)$ , jeżeli spełnione są następujące aksjomaty:
\begin{enumerate}[label=(G$_{\arabic*}$), leftmargin=*]
\item działanie  $\circ$  jest łączne,
\item działanie  $\circ$  posiada element neutralny,
\item każdy element posiada element odwrotny.
\end{enumerate}
\end{definicja}
```

**Definicja 1.6.** Grupą nazywamy parę  $(G, \circ)$ , jeżeli spełnione są następujące aksjomaty:

- $(G_1)$  działanie  $\circ$  jest łączne,
- $(G_2)$  działanie  $\circ$  posiada element neutralny,
- $(G_3)$  każdy element posiada element odwrotny.

Wyliczenia nienumerowane tworzymy następująco:

```
\begin{itemize}
  \item Jeżeli  $2|n$ , to  $n$  jest liczbą parzystą.
  \item Liczb parzystych jest nieskończenie wiele.
  \item Istnieje dokładnie jedna parzysta liczba pierwsza.
\end{itemize}
```

- Jeżeli  $2|n$ , to  $n$  jest liczbą parzystą.
- Liczb parzystych jest nieskończenie wiele.
- Istnieje dokładnie jedna parzysta liczba pierwsza.

## 1.8. Obrazki

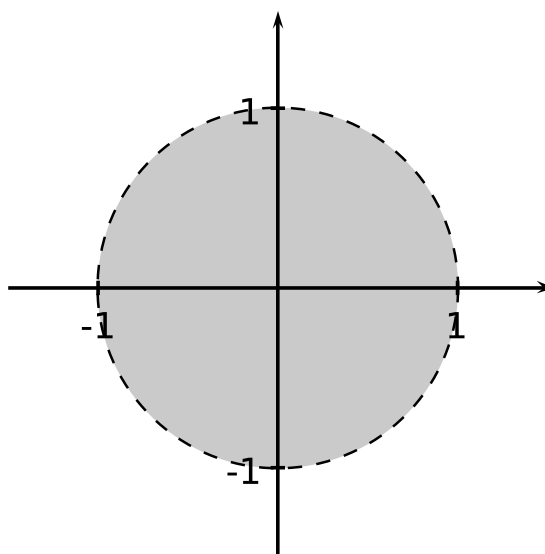
Temat umieszczenia obrazków w  $\text{\LaTeX}$ u jest zagadnieniem troszkę skomplikowanym, szczególnie jeżeli chcemy na obrazkach umieścić formuły i symbole matematyczne. Istnieją wyspecjalizowane pakiety do tworzenia obrazków: `pstricks` oraz `tikz`. Jednak ich użycie może wymagać sporo dodatkowej nauki.

Do utworzenia obrazków może służyć dowolny program graficzny z możliwością eksportu do formatu `*.eps` lub `*.pdf`. Polecane programy posiadające możliwością eksportu do wspomnianych formatów i umieszczania formuł  $\text{\TeX}$ owych:

- L<sup>A</sup>T<sub>E</sub>XDraw
- Inkscape

Po tworzeniu obrazków możemy je dołączyć w taki sposób:

```
\begin{figure*}[!h]
```



Rys. 1.1. Opis pod obrazkiem. Może być bardzo długi i zawierać wiele istotnych informacji. Może mówić co przedstawia obrazek, podawać parametry, wzory itp.

```

% wyśrodkowanie zawartości pola obrazka
\begin{center}
% okienko skalujące:
% pierwszy argument szerokość, drugi wysokość,
% jeden z nich może być zastąpiony ! - zachowanie proporcji obrazka
% w taki sposób możemy skalować także inne obiekty np. tekst
\resizebox{0.5\textwidth}{!}{
% wstawienie obrazka
\includegraphics{Rysunki/KulaEuklidesowa}
}
% opis obrazka
\caption[Skrócony opis do spisu obrazków]{Opis pod obrazkiem. Może być bardzo długi
i zawierać wiele istotnych informacji. Może mówić co przedstawia obrazek, podawać parametry, wz-
itp.}
% etykieta
\label{etykieta_obrazka}
\end{center}
\end{figure*}

```

Proponowanym rozwiązaniem jest także umieszczenie obrazków w odrębnym katalogu znajdującym się w głównym folderze pracy dyplomowej. Przykładowo w niniejszym szablonie utworzono katalog rysunki, z którego dołączyć można obrazek za pomocą standardowego sposobu pokazanego powyżej, dodając ścieżkę dostępu do obrazka np. :

```
\includegraphics[scale=0.5]{rysunki/obrazek1.jpg}
```

Do obrazków możemy odwoływać się używając konstrukcji `\label – \ref`, np.: Rysunek 1.1 przedstawia kulę w metryce euklidesowej.

Rysunek `\ref{kula_eulidesowa}` przedstawia kulę w metryce euklidesowej.

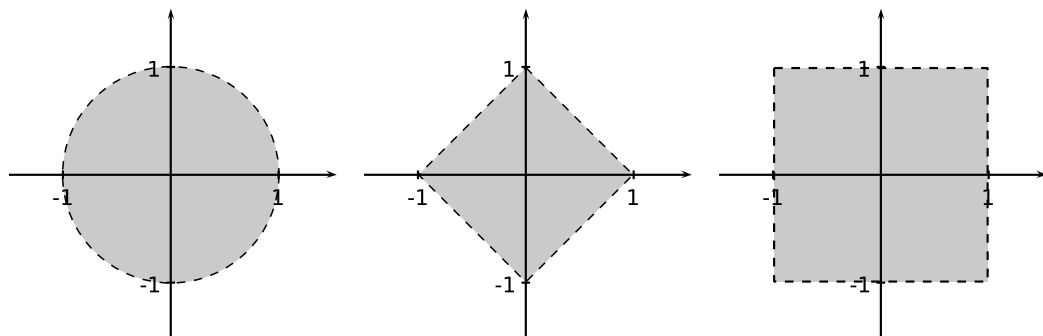
Spis obrazków dołączamy do pracy za pomocą polecenia `\listoffigures`. Sporym problemem może być umieszczenie obrazka w konkretnym miejscu na stronie. Jeżeli  $\LaTeX$  nie potrafi umieścić obrazka tam gdzie chcemy, to może go przenieść na następną stronę lub na koniec pliku. Prostim rozwiązaniem jest zmiana wielkości obrazka. W tym wypadku najlepiej spytać Google. Ogólna porada jest taka: pozycjonowanie obrazków należy wykonać dopiero po napisaniu całego tekstu, gdyż dodanie jednej linii tekstu może całkowicie zmienić pozycję obrazka.

Możliwe jest także umieszczenie kilku obrazków na jednym rysunku.

```

\begin{figure*}[!b]
\centering
\subfloat[$X=\mathbb{R}^2$, metryka taksówkowa]
{
\resizebox{0.3\textwidth}{!}{
\includegraphics{Rysunki/KulaEuklidesowa}
}
}
\subfloat[$X=\mathbb{R}^2$, metryka taksówkowa]

```



(a)  $X = \mathbb{R}^2$ , metryka taksówkowa (b)  $X = \mathbb{R}^2$ , metryka taksówkowa (c)  $X = \mathbb{R}^2$ , metryka maksimum

Rys. 1.2. Kule w różnych metrykach

```
{
  \resizebox{0.3\textwidth}{!}{
    \includegraphics{Rysunki/KulaTaksowkowa}
  }
}
\subfloat[ $X = \mathbb{R}^2$ , metryka maksimum]
{
  \resizebox{0.3\textwidth}{!}{
    \includegraphics{Rysunki/KulaMaksimum}
  }
}
\caption{Kule w różnych metrykach}
\end{figure*}
```

### 1.9. Dodawanie listingu kodu

W razie potrzeby zamieszczenia listingu kodu omawianego w pracy dyplomowej lub krótkich fragmentów tzw. pseudokodu, można wykorzystać pakiet `listings`. Pakiet ten zawarto w dołączonym do szablonu pakiecie `listing_schemat.sty` który znajduje się w głównym katalogu. Pakiet ten dodaje się przez odkomutowanie poniższego polecenia na początku pliku głównego **SzablonPracyPG\_main.tex**:

```
%----- Dodatkowe pakiety -----
\usepackage{listing_schemat}
%-----
```

Przykład zamieszczenia listingu kodu w pracy przedstawiono poniżej.

```
\lstset{style=python}

\begin{lstlisting}[caption={Przykładowy listing w języku Python}, label=python]
%%%%%% KOD ŹRÓDŁOWY %%%%%%%%%
#importing the time module
import time
#welcoming the user
name = raw_input("What is your name? ")
```



```

print "Hello, " + name, "Time to play hangman!"
#wait for 1 second
time.sleep(1)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
\end{lstlisting}

```

Po skompilowaniu, otrzymuje się listing w postaci przedstawionej poniżej.

```

1 #importing the time module
2 import time
3 #welcoming the user
4 name = raw_input("What is your name? ")
5 print "Hello, " + name, "Time to play hangman!"
6 #wait for 1 second
7 time.sleep(1)

```

Listing 1.1: Przykładowy listing w języku Python

Wstawienie listingu rozpoczyna się poleceniem `\lstset{style=NAZWA_JĘZYKA}`, które ustala określony sposób formatowania listingu zależnie od wybranego języka. Nazwę języka należy wpisać małymi literami. Opracowane formaty języków znajdują się w pliku pakietu **listing\_schemat**. Następnie do polecenia `\begin{listings}` podaje się argumenty wejściowe w kwadratowych nawiasach, co przedstawione zostało poniżej.

```

\begin{lstlisting}[caption={Podpis wstawianego listingu}, label=Odnosnik do listingu]
\end{lstlisting}

```

Podanie wymienionych argumentów wejściowych umożliwia podpisywanie poszczególnych listingów oraz odwoływanie się do nich w tekście przy użyciu polecenia `\ref{}`.

Istnieje także możliwość dołączania kodu źródłowego, który został zapisany w osobnym pliku. W tym celu należy użyć poniższego polecenia.

```

\begin{lstlisting}[caption={Podpis wstawianego listingu}, label=Odnosnik do listingu]
\lstinputlisting[language=Python, firstline=10, lastline=20]{plik_zrodlowy.py}
\end{lstlisting}

```

W pliku pakietu **listing\_schemat.sty** można zdefiniować własny styl formatowania listingu, zgodny z używanym językiem programowania. Parametry stylu formatowania zostały opisane w pliku, co umożliwia również ewentualną zmianę kolorów składni kodu lub innych atrybutów.

Do pakietu dołączono również środowisko **tikz**, które umożliwia generowanie przejrzystych schematów blokowych, grafów, diagramów oraz wykresów. Tworzenie diagramów poprzez opracowanie kodu w edytorze tekstu może być uciążliwe, jednak istnieją graficzne generatory kodu np. **TikzEdt** dostępny pod adresem: <http://www.tikzedt.org/>.

## 1.10. Dodawanie bibliografii

Bibliografię dodajemy za pomocą następujących poleceń

```

\bibliographystyle{plain} % styl bibliografii
\begin{thebibliography}{3} % początek środowiska
\addcontentsline{toc}{chapter}{Bibliografia} % dodaje bibliografię do spisu treści
\small % spisy i bibliografie składamy mniejszym stopniem pisma

```

```

% przykładowy wpis
    \bibitem{Duda}          % \bibitem{etykieta}
A. Duda, \emph{Wprowadzenie do topologii}, PWN, Warszawa 1986
% następna pozycja
    \bibitem{EngeSiek}
R. Engelking, K. Sieklucki, \emph{Geometria i topologia. Część II. Topologia}, PWN, Warszawa 19
% następna pozycja
    \bibitem{Patk}
H. Patkowska, \emph{Wstęp do topologii}, PWN, Warszawa 1979
% następna pozycja
    \bibitem{Siek}
K. Sieklucki, \emph{Geometria i topologia. Część I. Geometria}, PWN, Warszawa 1979
% następna pozycja
    \bibitem{Rutkowski}
Rutkowski J., \emph{Algebra Abstrakcyjna w zadaniach}, Wydawnictwo Naukowe PWN, Warszawa 2005
\end{thebibliography}          % koniec środowiska

```

Do pozycji bibliograficznych możemy odwoływać się w tekście korzystając z polecenia `\cite`, np.: Głównym źródłem jest książka A. Dudy [1]. Przykłady podane w tym rozdziale pochodzą z książki [5].

Głównym źródłem jest książka A. Dudy `\cite{Duda}`.

Przykłady podane w tym rozdziale pochodzą z książki `\cite{Rutkowski}`.

### 1.11. Uwagi o typografii

W polskiej typografii (i nie tylko w polskiej) przyjęło się stosować następujące zasady:

- na końcu wiersza nie mogą zostać słowa jednoliterowe (dwu-, trzy-) (tzw. sierota)
- akapit nie może kończyć się bardzo krótkim wierszem: pojedyncze słowo, przeniesiona część słowa lub krótkich słów, orientacyjnie, mniej niż 3 długości wcięcia akapitowego (tzw. wdowa) `<- wdowa`
- strona nie może kończyć się pojedynczym wierszem nowego akapitu (tzw. szewc)
- strona nie może zaczynać się ostatnim wierszem poprzedniego akapitu (tzw. bękart)
- unikać podwójnych wyróżnień, nie należy wyróżnianego fragmentu tekstu wyróżniać więcej niż jedną metodą naraz (np.: **podwójne wyróżnienie**)

[https://pl.wikipedia.org/wiki/Zasada\\_unikania\\_podwójnych\\_wyróżnień](https://pl.wikipedia.org/wiki/Zasada_unikania_podwójnych_wyróżnień)

Niestosowanie się do tych zasad nie umniejsza wartości pracy, jednak zmniejsza jej estetykę a w przypadku pojedynczych wierszy utrudnia czytanie. W tym dokumencie można znaleźć przykłady takich sytuacji.

Aby usunąć wspomniane błędy zwykle wystarczy zmienić szyk zdania lub użyć innego słowa. W ostateczności(!) można dodać pusty wiersz. Aby usunąć pojedyncze litery na końcu wiersza można dodać tzw. twardą spację, umieszczając pomiędzy wyrazami znak tyldy `~`.

**Przykład 1.7.** długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i

**Przykład 1.8.** długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i długiesłowo i krótkie i

Podobnie jak w przypadku obrazków, poprawki typograficzne powinny być wprowadzane po zakończeniu pisanie pracy, gdyż nawet najmniejsza zmiana w poprzednim wierszu może skutkować poważnymi przesunięciami całego tekstu.

Jako ciekawostka: norma PN-83/P-55366 (nieobowiązująca od kilku lat) podaje serię wskazówek i zaleceń typograficznych.

## 2. UWAGI OD STRONY TECHNICZNEJ

### 2.1. Opcje

1. `strict` – domyślnie, klasa stara się jak najściślej wypełniać zalecenia
2. `nostrict` – drobne modyfikacje typograficzne
  - zmniejszenie wcięcia akapitowego z 1.25cm na 1.5em

### 2.2. Wymagane pakiety

Lista pakietów, które są wymagane do kompilacji (większość z nich jest zapewne zainstalowana domyślnie)

1. `polski` – polonizacja  $\TeX$ 'a
2. `fontenc` – kodowanie znaków
3. `inputenc` – kodowanie znaków
4. `helvet` – wybiera font podobny do Arial
5. `geometry` – ustawienie marginesów
6. `indentfirst` – wcięcie pierwszego akapitu
7. `fancyhdr` – paginacja
8. `titlesec` – tytułaria
9. `titletoc` – formatowanie spisu treści
10. `enumitem` – wyliczenia numerowane i nienumerowane
11. `amsmath`, `amssymb`, `amsthm` – standardowe pakiety matematyczne
12. `graphicx` – dołączanie obrazków
13. `subfig` – wiele obrazków na jednym rysunku
14. `caption` – format podpisu pod obrazkiem
15. `tikz` – generowanie schematów blokowych i innych rysunków
16. `listings` – umieszczanie listingów kodu w pracy

### 2.3. Fonty

Wymagany fontem jest Arial. Ponieważ taki font nie jest łatwo dostępny w  $\LaTeX$ 'u więc korzystamy z fonta zastępczego w pakiecie `helv`. Wymagany font matematyczny nie został podany. Używamy zatem fontu z pakietu `mathpazo`.

```
\usepackage{helvet}
\usepackage{mathpazo}
\renewcommand{\familydefault}{\sfdefault}
```

Inną wersję fontu bezszeryfowego można uzyskać poprzez zrezygnowanie z pakietu `helv`.

Przykładowy:  $\sin(x) + ay^2$ .

$$\sin(x) + ay^2$$

## WYKAZ LITERATURY

- [1] A. Duda, *Wprowadzenie do topologii*, PWN, Warszawa 1986
- [2] R. Engelking, K. Sieklucki, *Geometria i topologia. Część II. Topologia*, PWN, Warszawa 1980
- [3] H. Patkowska, *Wstęp do topologii*, PWN, Warszawa 1979
- [4] K. Sieklucki, *Geometria i topologia. Część I. Geometria*, PWN, Warszawa 1979
- [5] Rutkowski J., *Algebra Abstrakcyjna w zadaniach*, Wydawnictwo Naukowe PWN, Warszawa 2005

## WYKAZ RYSUNKÓW

1.1	Skrócony opis do spisu obrazków . . . . .	6
1.2	Kule w różnych metrykach . . . . .	8

## WYKAZ TABEL