

POLITECHNIKA WROCŁAWSKA
WYDZIAŁ ELEKTRONIKI

KIERUNEK: INFORMATYKA
SPECJALNOŚĆ: INŻYNIERIA SYSTEMÓW INFORMATYCZNYCH

PRACA DYPLOMOWA
INŻYNIERSKA

Szablon pracy dyplomowej
inżynierskiej/magisterskiej, wersja 0.6

Engineering/master thesis template, version 0.6

AUTOR:

Imię Nazwisko

PROWADZĄCY PRACĘ:

tytuł, Imię Nazwisko, Jednostka

Opracował: Tomasz Kubik <tomasz.kubik@pwr.edu.pl>
Data: maj 2021



Tekst zawarty w niniejszym szablonie jest udostępniany na licencji Creative Commons: *Uznanie autorstwa – Użycie niekomercyjne – Na tych samych warunkach, 3.0 Polska*, Wrocław 2021. Oznacza to, że wszystkie przekazane treści można kopiować i wykorzystywać do celów niekomercyjnych, a także tworzyć na ich podstawie utwory zależne pod warunkiem podania autora i nazwy licencjodawcy oraz udzielania na utwory zależne takiej samej licencji. Tekst licencji jest dostępny pod adresem: <http://creativecommons.org/licenses/by-nc-sa/3.0/pl/>. Podczas redakcji pracy dyplomowej stronę tę można usunąć. Licencja dotyczy bowiem zredagowanego opisu, a nie samego latexowego szablonu. Latexowy szablon można wykorzystywać bez wzmiankowania o jego autorze.

Streszczenie

Streszczenie w języku polskim powinno zmieścić się na połowie strony (drugą połowę powinien zająć abstract w języku angielskim).

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

Słowa kluczowe: raz, dwa, trzy, cztery

Abstract

Streszczenie in Polish should fit on the half of the page (the other half should be covered by the abstract in English).

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

Keywords: one, two, three, four

Spis treści

1. Wstęp	9
1.1. Wprowadzenie	9
2. Praca z szablonem	11
2.1. Struktura projektu	11
2.2. Kodowanie znaków	12
2.3. Kompilacja szablonu	13
2.4. Sprawdzanie poprawności tekstu	14
2.5. Wersjonowanie	15
3. Zalecenia dotyczące formatowania	17
3.1. Rozmiar i układ treści na stronach dokumentu	17
3.2. Strona tytułowa	17
3.3. Krój i wielkość czcionek	20
3.4. Formatowanie bloków tekstu	21
3.5. Opisy tabel i rysunków	23
3.6. Przypisy dolne	23
3.7. Formatowanie spisu treści	23
3.8. Formatowanie list wyliczeniowych i wypunktowań	24
3.9. Wzory matematyczne	25
4. Redakcja pracy	26
4.1. Układ pracy	26
4.2. Styl	28
5. Uwagi techniczne	29
5.1. Zasady redakcji	29
5.2. Rysunki	30
5.3. Wstawianie kodu źródłowego	33
5.4. Wykaz literatury oraz cytowania	35
5.5. Indeks rzeczowy	36
5.6. Inne uwagi	37
6. Podsumowanie	39
6.1. Sekcja poziomu 1	39
6.1.1. Sekcja poziomu 2	39
6.2. Sekcja poziomu 1	39
Literatura	40
A. Instrukcja wdrożeniowa	41
B. Opis załączonej płyty CD/DVD	42

Spis rysunków

3.1. Układ strony nieparzystej dla dokumentu klasy memoir	18
3.2. Rzeczywisty układ strony nieparzystej w tym dokumencie	19
3.3. Oficjalny szablon strony tytułowej pracy dyplomowej, zamieszczony w dokumencie „System Identyfikacji Wizualnej Wrocław, sierpień 2016” do pobrania ze strony http://pwr.edu.pl/uczelnia/o-politechnice/materialy-promocyjne/logotyp [dostęp dnia 07.12.2016]	20
3.4. Parametry sterujące wielkościami odstępów na stronie z tytułem rozdziału	22
3.5. Kontrola ustawień odległości w tytułach kolejnych sekcji	22
3.6. Parametry sterujące przypisami dolnymi	24
3.7. Parametryzacja wyglądu spisu treści	24
3.8. Parametryzacja list wyliczeniowych i wypunktowań	25
5.1. Dwa znaki kanji – giri	31
5.2. Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne	31
5.3. Przykład diagramu: a) złego, b) w miarę dobrego	32
5.4. Przykłady zrzutów z ekranu: a) zły (nieczytelny, zrobiony przy zbyt szerokim oknie, z niepotrzebnymi marginesami, niepotrzebnym paskiem menu), b) w miarę dobry (w miarę czytelny, zrobiony przy zawężonym oknie, byłoby wskazane jeszcze usunięcie z niego belecзки z faviconem (jeśli nic nie wnosi) oraz przycięcie od dołu (jeśli treści tam pokazywane nie są istotne))	33

Spis tabel

1.1. Wykaz zalecanych narzędzi do pracy z wykorzystaniem szablonu (na dzień 09.02.2021)	9
2.1. Pliki źródłowe szablonu oraz wyniki kompilacji	12
3.1. Zestawienie czcionek elementów podziału dokumentu, tekstu wiodącego, nagłówka i stopki oraz podpisów (Rozm. – rozmiar czcionki, Odst. – baselineskip)	21

Spis listingów

5.1.	Kod źródłowy przykładów wstawiania rysunków do pracy	30
5.2.	Initial HTTP Request	33
5.3.	Opis 1	34
5.4.	Opis 2	34
5.5.	Kontrakt na model wejściowy endpointu <code>/api/v1/chess/board/move</code>	35

Skróty

OGC (ang. *Open Geospatial Consortium*)
XML (ang. *eXtensible Markup Language*)
SOAP (ang. *Simple Object Access Protocol*)
WSDL (ang. *Web Services Description Language*)
UDDI (ang. *Universal Description Discovery and Integration*)
GIS (ang. *Geographical Information System*)
SDI (ang. *Spatial Data Infrastructure*)
ISO (ang. *International Standards Organization*)
WMS (ang. *Web Map Service*)
WFS (ang. *Web Feature Service*)
WPS (ang. *Web Processing Service*)
GML (ang. *Geography Markup Language*)
SRG (ang. *Seeded Region Growing*)
SOA (ang. *Service Oriented Architecture*)
IT (ang. *Information Technology*)

Rozdział 1

Wstęp

1.1. Wprowadzenie

Niniejszy dokument powstał z myślą o ujednoliceniu sposobu redagowania prac dyplomowych. Jego źródła mają pełnić rolę szablonu nowoedytowanej pracy, zaś prezentowana treść posłużyć ma jako zbiór zaleceń i uwag o charakterze technicznym (dotyczących takich zagadnień, jak na przykład: formatowanie tekstu, załączanie rysunków, układ strony) oraz stylistycznym (odnoszących się do stylu wypowiedzi, sposobów tworzenia referencji itp.).

W źródłach szablonu zamieszczono komentarze z uwagami pozwalającymi lepiej zrozumieć znaczenie używanych komend. Komentarze te nie są widoczne w pliku `Dokument.pdf`, powstającym w wyniku kompilacji szablonu. Aby zrozumieć, jak dobrze wykorzystać szablon, należy sięgnąć do tych komentarzy.

Szablon przygotowano do kompilacji narzędziem `pdflatex` w konfiguracji: `MiKTeX` (windowsowa dystrybucja `latexa`) + `TeXnicCenter` (środowisko do edycji i kompilacji projektów `latexowych`) + `SumatraPDF` (przeglądarka `pdfów` z nawigacją zwrotną) + `JabRef` (opcjonalny edytor bazy danych bibliograficznych). Wymieniony zestaw narzędzi jest zalecany do pracy w systemie Windows. Narzędzia te można pobrać ze stron internetowych, których adresy zamieszczono w tabeli 1.1.

Tab. 1.1: Wykaz zalecanych narzędzi do pracy z wykorzystaniem szablonu (na dzień 09.02.2021)

Narzędzie	Wersja	Opis	Adres
MiKTeX	21.1	Zalecana jest instalacja Basic MiKTeX 32 lub 64 bitowa. Brakujące pakiety będą się doinstalowywać podczas kompilacji projektu	http://miktex.org/download
TeXnicCenter	2.02	Można pobrać 32 lub 64 bitową wersję	http://www.texniccenter.org/download/
SumatraPDF	3.2	Można pobrać 32 lub 64 bitową wersję	http://www.sumatrapdfreader.org/download-free-pdf-viewer.html
JabRef	5.2	Rozwijane w JDK 15, ma własny instalator i wersję przenośną	http://www.fosshub.com/JabRef.html

Wspomniana nawigacja między `TeXnicCenter` a `SumatraPDF` polega na przełączaniu się pomiędzy tymi środowiskami z zachowaniem kontekstu. Czyli edytując tekst w `TeXnicCenter` po kliknięciu na narzędziu podglądu można przeskoczyć do odpowiedniego miejsca w `pdfie` wyświetlanym przez `SumatraPDF`, podwójne kliknięcie w `pdfie` widocznym w `SumatraPDF` ustawi kursor

we właściwym akapicie w edytorze tekstu TeXnicCenter. O konfiguracji obu narzędzi do takiej współpracy napisano na stronie <http://tex.stackexchange.com/questions/116981/how-to-configure-texniccenter-2-0-with-sumatra-2013-2014-2015-version> (w sieci można znaleźć również inne materiały na ten temat).

Szablon można łatwo dostosować do innych narzędzi i środowisk (jak np. do multiplatformowego TexStudio oraz różnych dystrybucji systemu LaTeX). Dostosowanie to może polegać na zmianie kodowania plików, pominięciu pliku projektu `Dyplom.tcp` (patrz następny rozdział) i korekcie deklaracji kodowania znaków w dokumencie głównym (co opisano dalej).

Można go też zaadoptować do potrzeb pracy z wykorzystaniem latexowych edytorów i kompilatorów działających w trybie on-line. Takim narzędziem jest Overleaf (<https://www.overleaf.com/>). Choć ma niewątpliwie wiele zalet (w tym możliwość edycji przez wielu użytkowników jednocześnie), to podczas edycji długich dokumentów narzędzie to przegrywa ze wspomnianymi zintegrowanymi środowiskami TeXnicCenter czy TexStudio. Overleaf nie wyświetla struktury dokumentu (a jedynie listę plików), nie umożliwia wyszukiwania we wszystkich plikach projektowych (nie ma opcji `Find in Files ...`, która bardzo się przydaje do wyszukiwania wystąpień pewnych sentencji czy komend). Ponadto choć istnieje możliwość włączenia wersjonowania w Overleaf (w powiązaniu z jakimś gitowym repozytorium), to wersjonowanie to nie działa perfekcyjnie.

Dlatego zaleca się korzystanie ze środowisk zintegrowanych zainstalowanych lokalnie, czemu towarzyszyć ma wersjonowanie z pomocą wybranego zdalnego repozytorium (gitlab, github, bitbucket).

Rozdział 2

Praca z szablonem

2.1. Struktura projektu

Pisząc pracę w systemie LaTeX zwykle przyjmuje się jakąś konwencję co do nazewnictwa tworzonych plików, ich położenia oraz powiązań. Przygotowując niniejszy szablon założono, że projekt będzie się składał z pliku głównego, plików z kodem kolejnych rozdziałów i dodatków (włączanych do kompilacji w dokumencie głównym), katalogów z plikami grafik (o nazwach wskazujących na rozdziały, w których grafiki te zostaną wstawione), pliku ze skrótami (opcjonalny), pliku z danymi bibliograficznymi (plik `dokumentacja.bib`). Taki „układ” zapewnia porządek oraz pozwala na selektywną kompilację rozdziałów.

Przyjętą konwencję da się opisać jak następuje:

- Plikiem głównym jest plik `Dyplom.tex`. To w nim znajdują się deklaracje wszystkich używanych stylów, definicje makr oraz ustawień, jak również polecenie `\begin{document}`.
- Teksty redagowane są w osobnych plikach. Pliki te zamieszczone są w katalogu głównym (tym samym, co plik `Dyplom.tex`).
- W pliku `streszczenie.tex` powinien pojawić się tekst streszczenia ze słowami kluczowymi (tekst ten oraz słowa kluczowe będzie można wykorzystać do wypełnienia formularzy pojawiających się podczas wysyłania pracy do analizy antyplagiatowej w systemie ASAP)
- Plik `skroty.tex` powinien zawierać wykaz użytych skrótów. Jeśli w pracy nie stosuje się skrótów czy akronimów lub ich liczba jest nieznaczna, wtedy należy zrezygnować z tego pliku.
- Tekst kolejnych rozdziałów powinien pojawić się w plikach o nazwach zawierających numery tych rozdziałów. Według przyjętej konwencji `rozdzial01.tex` to plik pierwszego rozdziału (ze Wstępem), `rozdzial02.tex` to plik z treścią drugiego rozdziału itd.
- Teksty dodatków mają być redagowane w osobnych plikach o nazwach zawierających literę dodatku. Pliki te, podobnie do plików z tekstem rozdziałów, zamieszczane są w katalogu głównym. I tak `dodatekA.tex` oraz `dodatekB.tex` to, odpowiednio, pliki z treścią dodatku A oraz dodatku B.
- Każdemu rozdziałowi i dodatkowi towarzyszy katalog przeznaczony do składowania dołączanych w nim grafik. I tak `rys01` to katalog na pliki z grafikami dołączanymi do rozdziału pierwszego, `rys02` to katalog na pliki z grafikami dołączanymi do rozdziału drugiego itd. Podobnie `rysA` to katalog na pliki z grafikami dołączanymi w dodatku A itd.
- W katalogu głównym zamieszczany jest plik `dokumentacja.bib` zawierający bazę danych bibliograficznych.
- Jeśli praca nad dokumentem odbywa się w TeXnicCenter, to środowisko to wymusza konieczność podania pliku projektu (plik projektu to coś na styl pliku z definicją solucji). Plikiem projektu, który umieszczono w szablonie jest `Dyplom.tcp`. Generalnie plik projektu

nie jest wymagany do latexowej kompilacji. Niemniej pozwala zapamiętać ustawienia środowiska (w tym ustawienia językowe potrzebne do sprawdzania poprawności wyrazów – patrz następny podrozdział).

Tab. 2.1: Pliki źródłowe szablonu oraz wyniki kompilacji

Źródła	Wyniki kompilacji
Dokument.tex - dokument główny	Dyplom.bbl
Dokument.tcp – plik projektu TeXnicCenter (opcjonalny)	Dyplom.blg
streszczenie.tex – plik streszczenia	Dyplom.ind
skroty.tex – plik ze skrótami	Dyplom.idx
rozdzial01.tex – plik rozdziału 01	Dyplom.lof
...	Dyplom.log
dodatekA.tex – plik dodatku A	Dyplom.lot
...	Dyplom.out
rys01 – katalog na rysunki do rozdziału 01	Dyplom.pdf – dokument wynikowy
- fig01.png – plik grafiki	Dyplom.synctex
- ...	Dyplom.toc
...	Dyplom.tps
rysA – katalog na rysunki do dodatku A	*.aux
- fig01.png – plik grafiki	Dyplom.synctex
- ...	
...	
dokumentacja.bib – plik danych bibliograficznych	
Dyplom.ist – plik ze stylem indeksu	
by-nc-sa.png – plik z ikonami CC	

2.2. Kodowanie znaków

System LaTeX obsługuje wielojęzyczność. Można tworzyć w nim dokumenty z tekstem zawierającym różne znaki diakrytyczne. Należy jednak zdawać sobie sprawę, w jaki sposób znaki te są obsługiwane. Otóż na kodowanie znaków należy patrzeć z dwóch perspektyw: perspektywy edytowania kodu latexowego oraz perspektywy kodowania dokumentu wynikowego i użytych czcionek.

Kod latexowy może być edytowany w dowolnym tekstowym edytorze. Zastosowane kodowanie znaków w tym edytorze musi być znane latexowi, inaczej kompilacja tego kodu się nie powiedzie. Informację o tym kodowaniu przekazuje się w opcjach pakietu `inputenc`. Niniejszy szablon przygotowany w systemie Windows, a latexowe źródła umieszczono w plikach ANSI z użyciem strony kodowej `cp1250`. Dlatego w poleceniu `\usepackage[cp1250]{inputenc}` jako opcję wpisano `cp1250`.

Szablon można wykorzystać również przy innych kodowaniach i w innych systemach. Jednak wtedy konieczna będzie korekta dokumentu `Dyplom.tex` odpowiednio do wybranego przypadku. Korekta ta polegać ma na zamianie polecenia `\usepackage[cp1250]{inputenc}` na polecenie `\usepackage[utf8]{inputenc}` oraz konwersji znaków i zmiany kodowania istniejących plików ze źródłem latexowego kodu (plików o rozszerzeniu `*.tex` oraz `*.bib`).

Kodowanie znaków jest istotne również przy edytowaniu bazy danych bibliograficznych (pliku `dokumentacja.bib`). Aby `bibtex` poprawnie interpretował polskie znaki plik

dokumentacja.bib powinien być zakodowany w ANSI, CR+LF (dla ustawień jak w szablonie). W szczególności, jeśli ominąć chce się problem kodowania, polskie znaki w bazie danych bibliograficznych można zastąpić odpowiednią notacją: \k{a} \c{k{e}} \l{} \n \o \s \z \.z \k{A} \c{E} \L{} \N \O \S \Z \.Z.

Samo kodowanie plików może być źródłem paru problemów. Chodzi o to, że użytkownicy pracujący z edytorami tekstów pod linuxem mogą generować pliki zakodowane w UTF-8 bez BOM (lub z BOM – co nie jest zalecane), a pod windowsem – pliki ANSI ze znakami ze strony kodowej cp1250. A z takimi plikami różne edytory różnie sobie radzą. W szczególności edytor TeXnicCenter podczas otwierania plików może potraktować jego zawartość jako UTF8 lub ANSI – prawdopodobnie interpretuje z jakim kodowaniem ma do czynienia na podstawie obecności w pliku znaków specjalnych. Bywa, że choć wszystko w TeXnicCenter wygląda OK, to jednak kompilacja latexowa „nie idzie”. Problemem mogą być właśnie pierwsze bajty, których nie widać w edytorze.

Do konwersji kodowania można użyć Notepad++ (jest tam opcja „konwertuj” - nie mylić z opcją „kodu”, która przekodowuje znaki, jednak nie zmienia sposobu kodowania pliku).

Jeśli chodzi o drugą perspektywę, tj. kodowanie znaków w dokumencie wynikowym, to sprawa jest bardziej skomplikowana. Wiąże się z nią zarządzanie czcionkami, definiowanie mapowania itp. Szablon przygotowano tak, by wynikowy dokument zawierał polskie znaki diakrytyczne, które nie są zlepkami literki i ogonka.

2.3. Kompilacja szablonu

Kompilację szablonu można uruchamiać na kilka różnych sposobów. Wszystko zależy od używanego systemu operacyjnego, zainstalowanej na nim dystrybucji latexa oraz dostępnych narzędzi. Zazwyczaj kompilację rozpoczyna się wydając polecenie z linii komend lub uruchamia się ją za pomocą narzędzi zintegrowanych środowisk.

Kompilacja z linii komend polega na uruchomieniu w katalogu, w którym rozpakowano źródła szablonu, następującego polecenia:

```
> pdflatex Dyplom.tex
```

gdzie `pdflatex` to nazwa kompilatora, zaś `Dyplom.tex` to nazwa głównego pliku redagowanej pracy. W przypadku korzystania ze środowiska TeXnicCenter należy otworzyć dostarczony w szablonie plik projektu `Dyplom.tcp`, a następnie uruchomić kompilację narzędziami dostępnymi w pasku narzędziowym.

Aby poprawnie wygenerowały się wszystkie referencje (spis treści, odwołania do tabel, rysunków, pozycji literaturowych, równań itd.) kompilację `pdflatex` należy wykonać dwukrotnie, a czasem nawet trzykrotnie. Wynika to z konieczności zapamiętywania wyników kompilacji i ich wykorzystywania w kolejnych przebiegach. Tak dzieje się przy generowaniu odwołań do pozycji literaturowych oraz tworzeniu wykazu literatury).

Wygenerowanie danych bibliograficznych zapewnia kompilacja `bibtex` uruchamiana po kompilacji `pdflatex`. Można to zrobić z linii komend:

```
> bibtex Dyplom
```

lub wybierając odpowiednią pozycję z paska narzędziowego wykorzystywanego środowiska. Po kompilacji za pomocą `bibtex` na dysku pojawi się plik `Dyplom.bbl`. Dopiero po kolejnych dwóch kompilacjach `pdflatex` dane z tego pliku zostaną odpowiednio przetworzone i zrenderowane w wygenerowanym dokumencie. Tak więc po każdym wstawieniu nowego cytowania w kodzie dokumentu uzyskanie poprawnego formatowania dokumentu wynikowego wymaga powtórzenia następującej sekwencji kroków kompilacji:

```
> pdflatex Document.tex
> bibtex Document
> latex Document.tex
> latex Document.tex
```

Szczegóły dotyczące przygotowania danych bibliograficznych oraz zastosowania cytowań przedstawiono w podrozdziale 5.4.

W głównym pliku zamieszczono polecenia pozwalające sterować procesem kompilacji poprzez włączanie bądź wyłączanie kodu źródłowego poszczególnych rozdziałów. Włączanie kodu do kompilacji zapewniają instrukcje `\include` oraz `\includeonly`. Pierwsza z nich pozwala włączyć do kompilacji kod wskazanego pliku (np. kodu źródłowego pierwszego rozdziału `\include{rozdzial01.tex}`). Druga, jeśli zostanie zastosowana, pozwala określić, które z plików zostaną skompilowane w całości (na przykład kod źródłowy pierwszego i drugiego rozdziału `\includeonly{rozdzial01.tex,rozdzial02.tex}`). Brak nazwy pliku na liście w poleceniu `\includeonly` przy jednoczesnym wystąpieniu jego nazwy w poleceniu `\include` oznacza, że w kompilacji zostaną uwzględnione referencje wygenerowane dla tego pliku wcześniej, sam zaś kod źródłowy pliku nie będzie kompilowany.

W szablonie wykorzystano klasę dokumentu `memoir` oraz wybrane pakiety. Podczas kompilacji szablonu w `MikTeX` wszelkie potrzebne pakiety zostaną zainstalowane automatycznie (jeśli `MikTeX` zainstalowano z opcją dynamicznej instalacji brakujących pakietów). W przypadku innych dystrybucji latexowych może okazać się, że pakiety te trzeba doinstalować ręcznie (np. pod linuxem z `TeXLive` trzeba doinstalować dodatkową zbiorczą paczkę, a jeśli ma się menadżera pakietów latexowych, to pakiety latexowe można instalować indywidualnie).

Jeśli w szablonie będzie wykorzystany indeks rzeczowy, kompilację źródeł trzeba będzie rozszerzyć o kroki potrzebne na wygenerowanie plików pośrednich `Dokument.idx` oraz `Dokument.ind` oraz dołączenia ich do finalnego dokumentu (podobnie jak to ma miejsce przy generowaniu wykazu literatury). Szczegóły dotyczące generowania indeksu rzeczowego opisano w podrozdziale 5.5.

2.4. Sprawdzanie poprawności tekstu

Większość środowisk ułatwiających pisanie latexowych dokumentów wspiera sprawdzenie poprawności tekstu (ang. *spell checking*). Wystarczy odpowiednio je skonfigurować. Niestety, proponowana przez narzędzia korekta nie jest genialna. Bazuje ona na prostym porównywaniu wyrazów (z końcówkami). Nie wbudowano w nią żadnej większej inteligencji. Tak więc proszę nie porównywać jej z korektą oferowaną w narzędziach MS Office (tam jest ona dużo bardziej zaawansowana).

TeXnicCenter korzysta ze słowników do pobrania ze strony [openoffice](https://extensions.openoffice.org/) (<https://extensions.openoffice.org/>). Aby sprawę uprościć słowniki dla języka polskiego (pliki `pl_PL.aff` oraz `pl_PL.dic`) dołączono do szablonu (są w katalogu `Dictionaries`). Pliki te należy umieścić w katalogu `C:`

Program Files

TeXnicCenter

`Dictionaries`, a w konfiguracji projektu (`Tools/Options/Spelling` należy wybrać `Language: pl`, `Dialect: PL`). Jeśli główny tekst pracy pisany jest w innym języku, to trzeba zmienić słownik.

Zaskakujące może być to, że *spell checker* w TeXnicCenter działa zarówno przy pracy na plikach UTF-8, jak i na plikach ANSI. Jeśli byłyby jakieś problemy ze słownikiem wynikające z kodowania znaków, wtedy słownik trzeba przekodować. To powinno pomóc.

2.5. Wersjonowanie

W trakcie edytowania pracy w systemie latex dobrą praktyką jest wersjonowanie tworzonego kodu. Do wersjonowania zaleca się wykorzystać system git. Opis sposobu pracy z tym systemem opisano w licznych tutorialach dostępnych w sieci. Szczególnie godnym polecenia zasobem jest strona domowa projektu <https://git-scm.com/>.

Zwykle pracę z gitem rozpoczyna się od utworzenia repozytorium zdalnego i lokalnego. Lokalne służy do bieżącej pracy, zdalne – do współpracy z innymi użytkownikami (z promotorem).

Po utworzeniu repozytorium lokalnego i jego gałęzi (czy będzie to master czy inna gałąź – wszystko zależy od ustaleń między zainteresowanymi) należy skopiować do niego wszystkie pliki dostarczone w szablonie, a po ich wstępnym przeredagowaniu należy je zaznaczyć do wersjonowania. Potem należy wysłać zmiany na repozytorium zdalne (możliwa jest też ścieżka odwrotna - można zacząć od zmian na repozytorium zdalnym, które pobrane będą do repozytorium lokalnego).

Podczas kompilowania projektu będą powstawały pliki pomocnicze. Plików tych nie należy wersjonować (zabierają niepotrzebnie miejsce, a przecież zawsze można je odtworzyć uruchamiając kompilację na źródłach). Git posiada mechanizm automatycznego odrzucania plików niepodlegających wersjonowaniu. Mechanizm ten bazuje na wykorzystaniu pliku konfiguracyjnego `.gitignore` zamieszczonego w katalogu głównym repozytorium. O szczegółach `.gitignore` można poczytać na stronie <https://git-scm.com/docs/gitignore>.

W sieci można znaleźć liczne propozycje plików konfiguracyjnych `.gitignore` dopasowanych do potrzeb latexowej kompilacji. Nie trzeba ich jednak szukać. Aby sprawę uprościć w szablonie zamieszczono specjalnie spreparowany taki plik. Zawiera on, między innymi, wpis mówiący o tym, by nie wersjonować pliku wynikowego `Dyplom.pdf`. Plik ten należy umieścić w repozytorium zaraz po jego utworzeniu.

Jeśli w repozytorium pojawiły się już jakieś zmiany zanim do niego wstawiono `.gitignore`, to wtedy należy wykonać kroki opisane na stronie: <https://stackoverflow.com/questions/38450276/force-git-to-update-gitignore/38451183>

```
> > > > You will have to clear the existing git cache first.
    git rm -r --cached .
> > > > Once you clear the existing cache, adds/stages all of the
    ↪ files in the current directory and commit
    git add .
    git commit -m "Suitable Message"
> > > >
```

Zalecany schemat współpracy dyplomanta z promotorem polega na wykonywaniu w kolejnych iteracjach następujących kroków:

- dyplomant edytuje wybraną część pracy, a po skończeniu edycji wrzuca zmiany do zdalnego repozytorium.
- dyplomant informuje promotora o zakończeniu etapu prac
- dyplomant może zacząć edycję kolejnego fragmentu pracy, a w tym czasie promotor może dokonać oceny/korekty zmian pobranych ze zdalnego repozytorium
- promotor wrzuca dokonane przez siebie zmiany do zdalnego repozytorium, informując o tym dyplomanta

Główna zasada tego schematu polega na niedoprowadzaniu do konfliktów (nadpisywanie się zmian). Jeśli jednak takie konflikty nastąpią, można je niwelować poprzez odpowiednie merge.

Do wzajemnego informowania można wykorzystać pocztę elektroniczną. Można też spróbować wdrożyć mechanizm zatwierdzania zmian (ang. *merge requests*). Można też umówić się

na sprawdzanie zawartości repozytorium zgodnie z jakimś przyjętym harmonogramem. Ważne, by wiadomo było obu stronom, na jakim schemacie współpracy mają bazować.

Dobłą praktyką jest też wstawianie w kod komentarzy. Przyjętą powszechnie konwencją jest rozpoczynanie komentarzy od:

- % TO DO: tekst zalecenia – jeśli jest to jakieś zalecenie promotora, czy też
- % DONE: tekst wyjaśnienia – jeśli jakieś zalecenie zostało wykonane przez dyplomanta.

Jako zdalne repozytorium można wykorzystać: github, bitbucket, gitlab (są to serwisy, które pozwalają zarządzać repozytoriami git). Dobłą praktyką jest też uruchomienie klientów git oferujących graficzny interfejs (jak SourceTree, GitCracken itp.). W narzędziach tych można zobaczyć natychmiast na czym polegały wprowadzone w repozytorium zmiany.

Rozdział 3

Zalecenia dotyczące formatowania

Większa część niniejszego rozdziału ma charakter informacyjny. Zamieszczone w nim informacje mają uzmysłowić czytelnikowi, jak wiele jest parametrów do ustawienia, by zredagowany dokument wyglądał „ładnie”. Parametry te poustawiano w szablonie. Wystarczy z niego skorzystać.

3.1. Rozmiar i układ treści na stronach dokumentu

Praca dyplomowa powinna być przygotowana do wydruku na papierze formatu A4 w orientacji pionowej. Marginesy na stronach parzystych i nieparzystych powinny być jednakowe i mieć następujące wartości: lewy = 25mm, prawy = 25mm, górny = 10mm, dolny = 15mm. Wielkość marginesów w szablonie sterowana jest parametrami przedstawionymi na rysunku 3.1. Margines dolny powinien być mierzony do linii bazowej tekstu stopki.

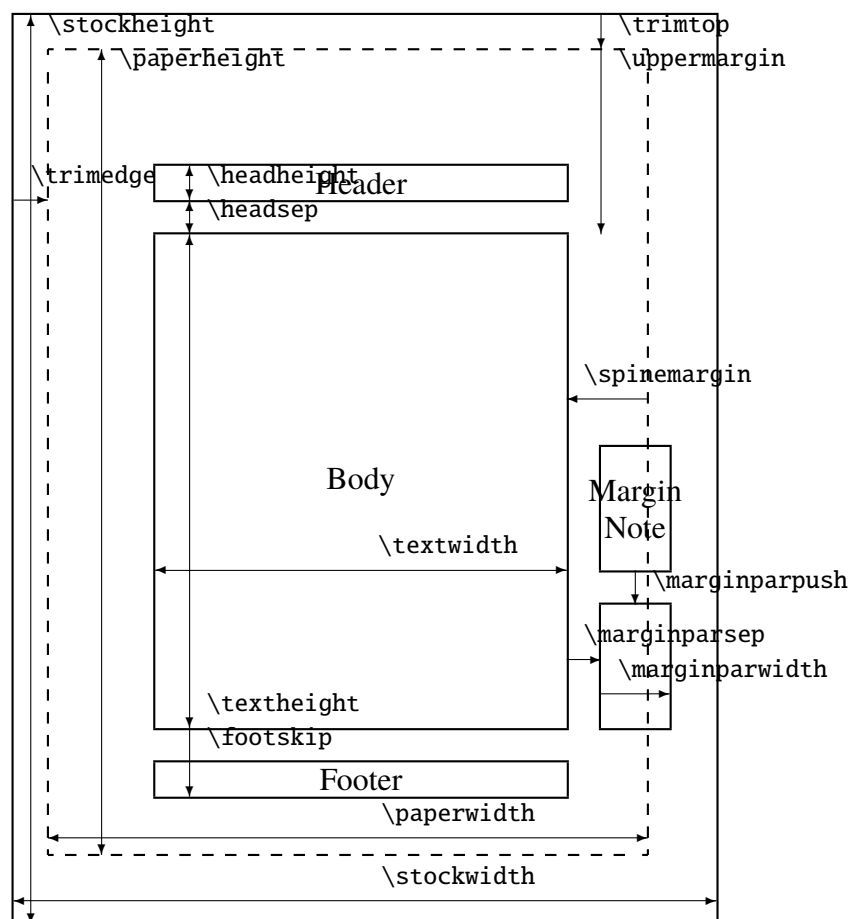
Rzeczywisty układ strony zastosowany w niniejszym dokumencie przedstawiono na rysunku 3.2. Lewy i prawy margines są takie same, więc strony parzyste i nieparzyste wyglądają podobnie, z dokładnością do umiejscowienia notatek marginesowych. Taki rezultat zapewniło zastosowanie poniższych komend.

```
\setlength{\headsep}{10pt}  
\setlength{\headheight}{13.6pt}  
\setlength{\footskip}{\headsep+\headheight}  
\setlength{\uppermargin}{\headheight+\headsep+1cm}  
\setlength{\textheight}{\paperheight-\uppermargin-\footskip-1.5cm}  
\setlength{\textwidth}{\paperwidth-5cm}  
\setlength{\spinewidth}{2.5cm}  
\setlength{\foremargin}{2.5cm}  
\setlength{\marginparsep}{2mm}  
\setlength{\marginparwidth}{2.3mm}  
\checkandfixthelayout[fixed]  
\linespread{1}  
\setlength{\parindent}{14.5pt}
```

3.2. Strona tytułowa

Według ogólnouczelnianych zaleceń (tj. logotypu Politechniki Wrocławskiej) strona tytułowa powinna być zredagowana z użyciem czcionki garamond. W oficjalnym wzorcu (patrz rysunek 3.3) nie rozróżniono, czy dotyczy on pracy inżynierskiej czy magisterskiej. Nie uwzględniono również miejsca na nazwę specjalności ani kierunku oraz zapomniano o nazwisku promotora, jednostce, dacie i ocenie. Za to określono położenie słów kluczowych i streszczenia.

Dashed lines represent the actual page size after trimming the stock.



Rys. 3.1: Układ strony nieparzystej dla dokumentu klasy memoir

Ponieważ brakujące dane pojawiały się we wzorcach stron tytułowych stosowanych w codziennej praktyce na Wydziałach, nie wiadomo do końca, czy oficjalny szablon należy stosować w 100 procentach. Dlatego w niniejszym dokumencie zastosowano własny wzorec strony tytułowej (używany od lat) oraz podano wymagania odnośnie wzorca z logotypu uczelnianego.

Wymagania co do wielkości znaków na stronie tytułowej są następujące:

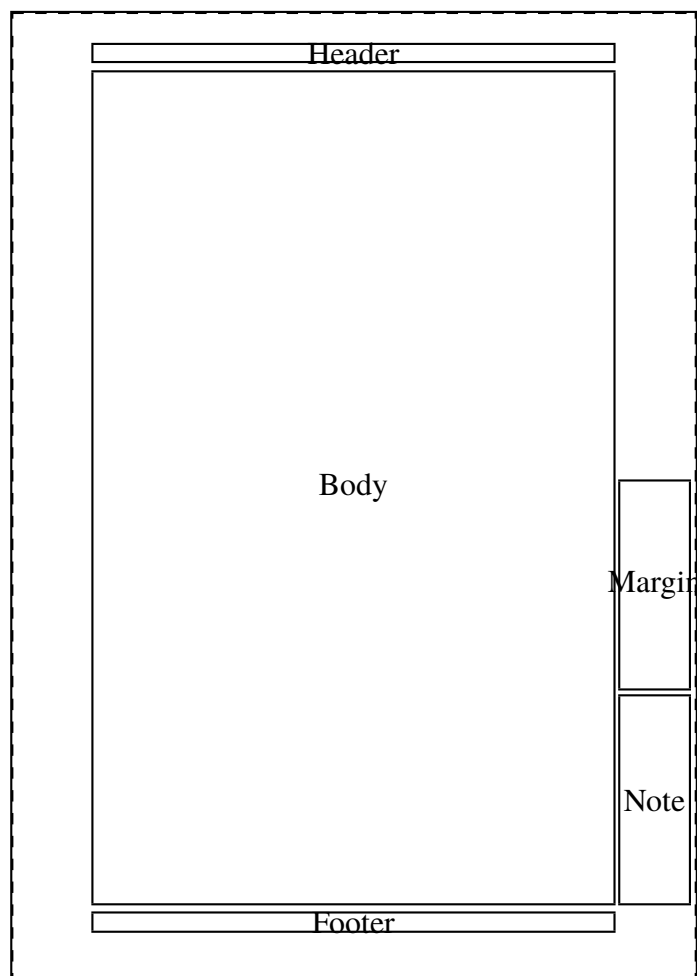
- według uczelnianego logotypu

Nazwa jednostki organizacyjnej: Garamond 16 pt
 Napis "PRACA DYPLMOWA INŻYNIERSKA": Garamond 32 pt
 Tytuł pracy: Garamond 16 pt
 Autor: Garamond 14 pt
 Słowa kluczowe: Garamond 12 pt
 Krótkie streszczenie: Garamond 10 pt

- według wzorca użytego w niniejszym dokumencie

POLITECHNIKA WROCŁAWSKA (Garamond 22pt 24pt)
 WYDZIAŁ ELEKTRONIKI (Garamond 22pt 24pt)
 KIERUNEK: JAKIŚ KIERUNEK (Garamond 14pt 16pt)
 SPECJALNOŚĆ: JAKAŚ SPECJALNOŚĆ (Garamond 14pt 16pt)
 PRACA DYPLMOWA (Garamond 24pt 26pt)
 INŻYNIERSKA (Garamond 24pt 26pt)
 Tytuł pracy w języku polskim (Garamond 16pt 18pt)
 Title in English (Garamond 16pt 18pt)
 AUTOR: (Garamond 16pt 18pt)

Dashed lines represent the actual page size after trimming the stock.



Lengths are to the nearest pt.

<code>\stockheight = 845pt</code>	<code>\stockwidth = 598pt</code>
<code>\pageheight = 845pt</code>	<code>\pagewidth = 598pt</code>
<code>\textheight = 727pt</code>	<code>\textwidth = 455pt</code>
<code>\trimtop = 0pt</code>	<code>\trimedged = 0pt</code>
<code>\uppermargin = 52pt</code>	<code>\spinemargin = 71pt</code>
<code>\headheight = 14pt</code>	<code>\headsep = 10pt</code>
<code>\footskip = 24pt</code>	<code>\marginparsep = 6pt</code>
<code>\marginparpush = 7pt</code>	<code>\columnsep = 10pt</code>
<code>\columnseprule = 0.0pt</code>	

Rys. 3.2: Rzeczywisty układ strony nieparzystej w tym dokumencie

Imię Nazwisko (Garamond 14pt 16pt)
 PROWADZĄCY PRACĘ: (Garamond 16pt 18pt)
 tytuł, Imię Nazwisko, Jednostka (Garamond 14pt 16pt)
 OCENA PRACY: (Garamond 16pt 18pt)
 WROCŁAW, 2015 (Garamond 16pt 18pt)

W szablonie zastosowano pakiet `ebgaramond`. Dostarcza on klonu czcionki `garamond`, jednak bez kształtu `slanted` i z pewnymi brakami. Na przykład zamiast literki „ł” w zbiorze `EBGaramond08` `Italic` renderuje się samo „l” (braku tego nie ma zbiór `EBGaramond12`). Zaletą pakietu w porównaniu do innych jest to, że generalnie dobrze obsługiwane są w nim polskie

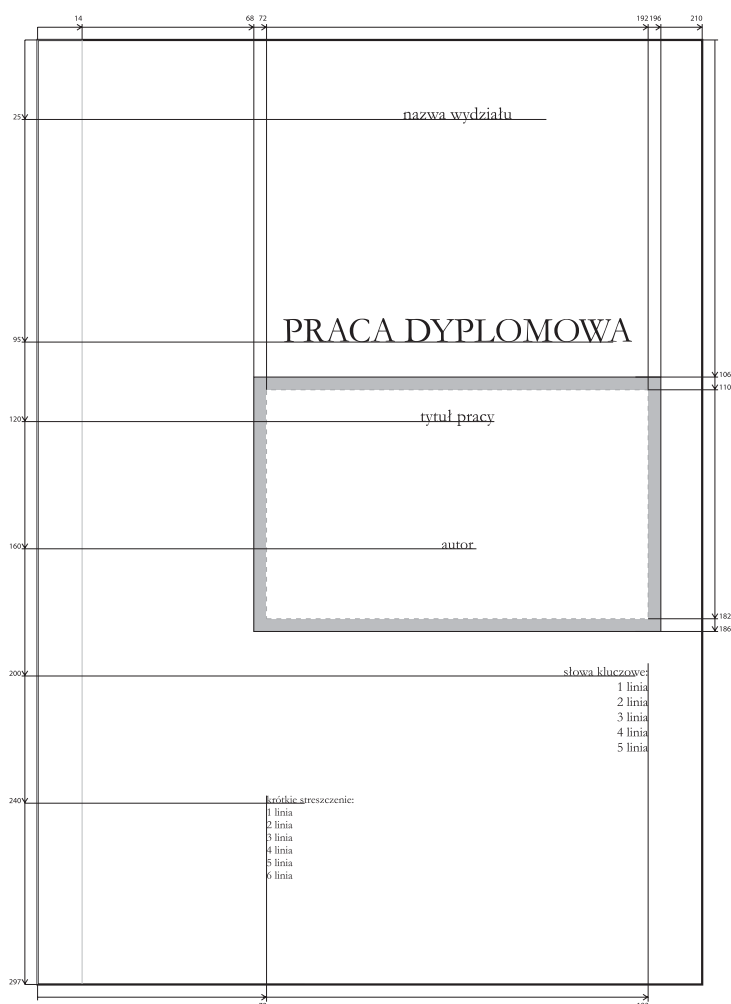
znaki oraz że pakiet ten można znaleźć w różnych dystrybucjach latexa (MikTeX instaluje go automatycznie).

3.3. Krój i wielkość czcionek

Główny tekst pracy powinien być zredagowany z wykorzystaniem czcionki Times, typ normalny, o wysokości 12pt, z odstępem między liniami równym 14.5pt. Istnieje możliwość zmiany odstępu między liniami za pomocą komendy `\linespread`, jednak zaleca się pozostawienie tego odstępu jak w niniejszym dokumencie (`\linespread{1}`). Wymagania odnośnie kroju pisma pozostałych elementów (nagłówków, stopek itp.) zamieszczono w tabeli 3.1.

W szablonie zastosowano czcionkę `texgyre-termes` (dostarcza ją pakiet `tgtermes`). Czcionka ta jest klonem czcionki Times, w którym obsługiwane jest środkowoeuropejskie kodowanie znaków (podobnie jak w przypadku czcionki `ebgaramond`, dzięki czemu polskie literki nie są zlepkami dwóch znaków lecz pojedynczymi znakami).

Wszelkie przykłady źródeł kodu (fragmenty programów, komendy linii poleceń), nazwy plików i uruchamianych programów powinny być pisane czcionką maszynową. W szablonie czcionką maszynową jest `tlx tt`. Czcionka ta obsługuje polskie znaki. Dostarcza ją pakiet



Rys. 3.3: Oficjalny szablon strony tytułowej pracy dyplomowej, zamieszczony w dokumencie „System Identyfikacji Wizualnej Wrocław, sierpień 2016” do pobrania ze strony <http://pwr.edu.pl/uczelnia/o-politechnice/materialy-promocyjne/logotyp> [dostęp dnia 07.12.2016]

`txfonts`, który należy wcześniej zainstalować (MiKTeX zainstaluje go automatycznie podczas pierwszej kompilacji szablonu).

Tab. 3.1: Zestawienie czcionek elementów podziału dokumentu, tekstu wiodącego, nagłówka i stopki oraz podpisów (Rozm. – rozmiar czcionki, Odst. – `baselineskip`)

Element	Przykład	Czcionka	Rozm.	Odst.
Nr rozdziału	Rozdział 1	<code>\huge \bfseries</code>	25pt	30pt
Tytuł rozdziału	Wstęp	<code>\Huge \bfseries</code>	30pt	37pt
Nr i tytuł sekcji	1.1. Wprowadzenie	<code>\Large \bfseries</code>	17pt	22pt
Nr i tytuł podsekcji	1.1.1. Cel szczegółowy	<code>\large \bfseries</code>	14.5pt	18pt
Tytuł podpodsekcji	Założenia	<code>\normalsize \bfseries</code>	12pt	14.5pt
Tytuł paragrafu	Podstawy Opis ...	<code>\normalsize \bfseries</code>	12pt	14.5pt
Tekst wiodący	Niniejszy dokument ...	<code>\normalsize</code>	12pt	14.5pt
Nagłówek strony	3.2. <i>Czcionka wiodąca</i> ...	<code>\small \itshape</code>	11pt	13.6pt
Stopka strony	Imię Nazwisko: ...	<code>\small</code>	11pt	13.6pt
Podpisy tabel	Tab. 3.1: Zestawienie ...	<code>\small</code>	11pt	13.6pt
Podpisy rysunków	Rys. 3.1: Oficjalny ...	<code>\small</code>	11pt	13.6pt

Jeśli w pracy zostaną użyte otoczenia matematyczne, to w dokumencie wynikowym pojawią się dodatkowe czcionki (domyślne latexowe czcionki do wyrażeń matematycznych). Dzięki zastosowaniu opcji `extrafontsizes` w klasie `memoir` nie dość, że otrzymuje się większe czcionki (30pt), to jeszcze zamiast `Computer Modern` do wzorów matematycznych jest stosowana czcionka `Latin Modern` (wywodząca się z `Computer Modern`). Stąd lista wszystkich użytych czcionek może być następująca:

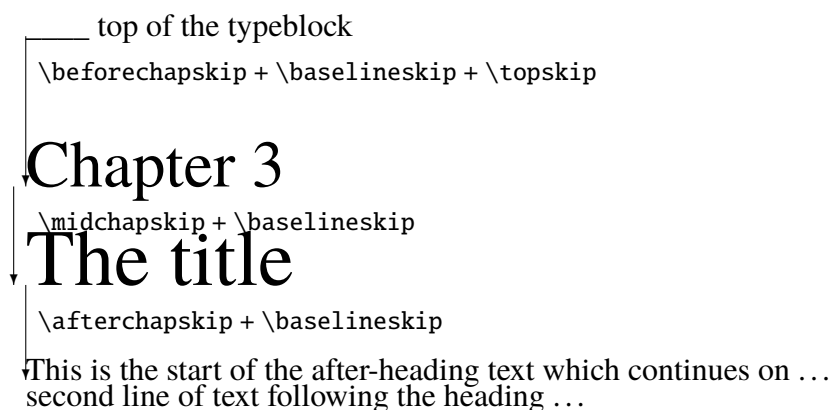
```
EBGaramond12-Regular
GaramondNo8-Reg-Norml
TeXGyreTermes-Regular-Normalna
TeXGyreTermes-Bold-Pogrubiona
TeXGyreTermes-Italic-Normalna
t1x11-Nomal
LMMathItalic12-Regular
LMMathSymbols10-Regular
LMMathExtension10-Regular
LMRoman8-Regular
```

Aby wykorzystać te czcionki poza systemem LaTeX, wystarczy pobrać je spod adresów (ważnych na dzień 1.04.2016): <https://www.ctan.org/tex-archive/fonts/cm/ps-type1/bakoma/ttf/?lang=en>, <http://www.gust.org.pl/projects/e-foundry/latin-modern>, <http://www.gust.org.pl/projects/e-foundry/tex-gyre>, <https://bitbucket.org/georgd/eb-garamond/downloads>, a następnie zainstalować w systemie. Dzięki temu można będzie np. edytować rysunki używając dokładnie tej samej czcionki, co czcionka użyta w dokumencie.

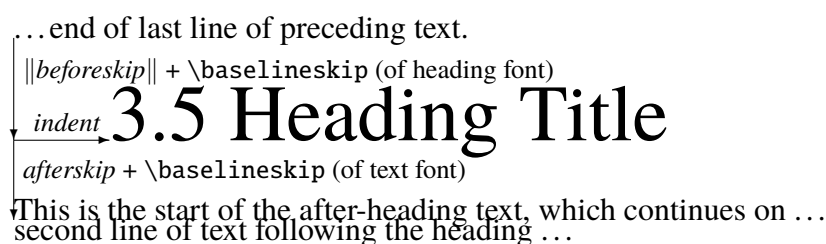
3.4. Formatowanie bloków tekstu

Każdy rozdział pracy powinien rozpoczynać się od nowej strony. Jej wygląd powinien być kontrolowany parametrami pokazanymi na rysunku 3.4. W niniejszym szablonie (dokument klasy `memoir` z opcją `[12pt]`) przyjęto następujące wartości tych parametrów:

- `\beforechapskip (50.0pt) + \baselineskip of \huge (30pt) + \topskip (12.0pt) = 92pt (3.246cm)`



Rys. 3.4: Parametry sterujące wielkościami odstępów na stronie z tytułem rozdziału



Rys. 3.5: Kontrola ustawień odległości w tytułach kolejnych sekcji

- $\backslash\text{midchapskip}$ (20.0pt) + $\backslash\text{baselineskip}$ of $\backslash\text{Huge}$ (37pt) = 57 pt (2.011cm)
- $\backslash\text{afterchapskip}$ (40.0pt) + $\backslash\text{baselineskip}$ of $\backslash\text{normalsize}$ (14.5pt) = 54.5pt (1.923cm)

Nieco kłopotów może sprawić dobre ustawienie na stronie tytułów nienumerowanych rozdziałów oraz list generowanych automatycznie (Skróty, Spis treści, Spis rysunków, Spis tabel, Indeks rzeczowy). W szablonie w tym celu zdefiniowano nowy styl rozdziału komendami jak niżej (w szablonie są to komendy zamarkowane)

```
\newlength{\linespace}
\setlength{\linespace}{-\beforechapskip-\topskip+\headheight+\topsep}
\makechapterstyle{noNumbered}{%
\renewcommand\chapterheadstart{\vspace*{\linespace}}
}
```

oraz dokonano przełączenia stylów rozdziałów komendami `\chapterstyle{nonumbered}` oraz `\chapterstyle{default}` podczas dołączania do dokumentu wymienionych nienumerowanych rozdziałów i list. Aby „podnieść do góry” tytuły nienumerowanych rozdziałów (gdy jest to rzeczywiście konieczne) wystarczy odmarkować wspomniane komendy.

Tytuły rozdziałów, sekcji, podsekcji itd. nie powinny kończyć się kropką. Odległości pomiędzy tekstem wiodącym a tytułem sekcji powinien być regulowany parametrami pokazanymi na rysunku 3.5. Rozmiar `\baselineskip` zależy od rozmiaru czcionki (zobacz tabela 3.1), zaś `beforeseckskip` i `secskip` od poziomu sekcji. W niniejszym szablonie przyjęto następujące wartości tych parametrów (są to wartości dobierane elastycznie podczas kompilacji):

- `indent` = 14.5pt
- `parskip` = 0.0pt
- `beforeseckskip` = -18.08334pt plus -5.16667pt minus -1.03331pt

- `aftersecskip = 11.88335pt plus 1.03331pt`
- `beforesubsecskip = -16.79167pt plus -5.16667pt minus -1.03331pt`
- `aftersubsecskip = 7.75pt plus 1.03331pt`
- `beforesubsubsecskip = -16.79167pt plus -5.16667pt minus -1.03331pt`
- `aftersubsubsecskip = 7.75pt plus 1.03331pt`

W szablonie obowiązują również następujące wartości parametrów odpowiedzialnych za odstępy pomiędzy pływającymi figurami, tekstami oraz tekstem i figurą:

- `floatsep = 12.0pt plus 2.0pt minus 2.0pt`
- `intextsep = 14.0pt plus 4.0pt minus 4.0pt`
- `textfloatsep = 20.0pt plus 2.0pt minus 4.0pt`

Pierwsza linia pierwszego akapitu w bloku (po tytule rozdziału, sekcji, podsekcji, podpodsekcji) nie może mieć wcięcia. Pierwsze linie w kolejnych akapitach już powinny mieć wcięcie równe 14.5pt. Tekst w akapitach powinien być wyrównany z obu stron.

Strony powinny być numerowane numeracją ciągłą (sekwencja arabskich cyfr). Numery stron powinny być umieszczone w ich stopkach (tj. tak jak w niniejszym dokumencie). Wyjątkiem są tutaj pierwsze strony rozdziałów oraz strona tytułowa – na nich numery nie powinny się pojawić.

3.5. Opisy tabel i rysunków

Podpisy powinny być umieszczane pod rysunkami lub nad tabelami wraz z etykietą składającą się ze skrótu Rys. lub Tab. oraz numeru. Podpisy te nie powinny mieć końcowej kropki. Numery występujące w podpisach powinny zaczynać się numerem rozdziału, po którym następuje kolejny numer rysunku lub tabeli w obrębie rozdziału. Etykieta powinna kończyć się dwukropkiem, po którym następuje tekst podpisu. Numer rozdziału powinien być rozdzielony kropką od kolejnego numeru w rysunku bądź tabeli w rozdziale (liczniki tabel i rysunków są rozłączne). Należy pamiętać o tym, żeby w całej pracy tabele miały podobny wygląd (rodzaj czcionki, ewentualne pogrubienia w nagłówku itp.).

3.6. Przypisy dolne

Istnieje możliwość zamieszczania przypisów na dole strony, choć nie jest to zalecane (przykładowo ¹). Sposób parametryzowania ich wyglądu pokazano na rysunku 3.6. W szablonie wykorzystano następujące, domyślne wartości tych parametrów:

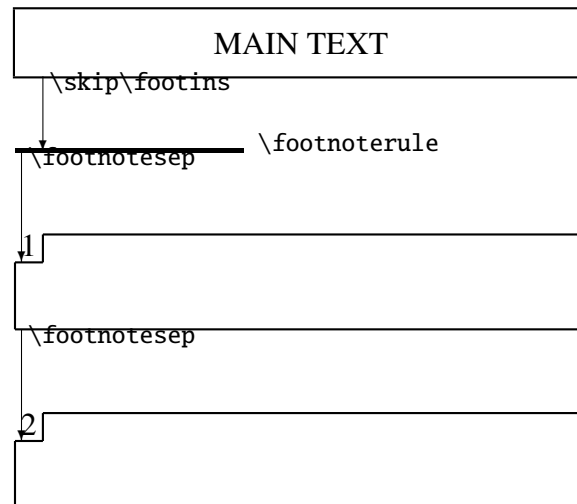
```
\footins = 12pt \footnotesep = 8pt
\baselineskip = 10pt note separation = 40pt
rule thickness = 0.4pt
rule length = 0.25 times the \textwidth
```

3.7. Formatowanie spisu treści

W klasie memoir istnieją komendy pozwalające dość dobrze zarządzać wyglądem spisu treści. Na rysunku 3.7 pokazano, za pomocą jakich parametrów można wpływać na finalną jego postać. W szablonie wykorzystano następujące, domyślne ich wartości:

```
indent = 18pt
numwidth = 28pt
\@tocrmarg = 31pt
```

¹Tekst przypisu



Rys. 3.6: Parametry sterujące przypisami dolnymi

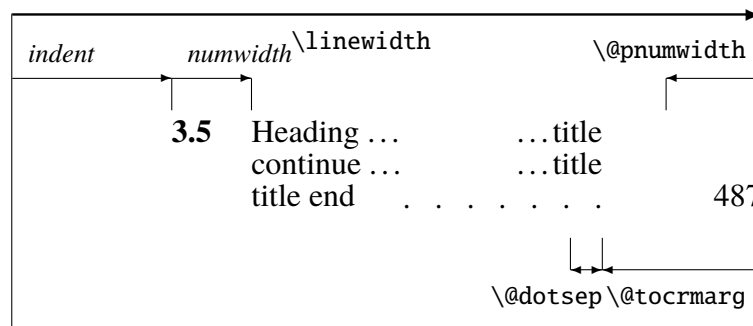
```
\@pnumwidth = 19pt
\@dotsep = 4.5
```

3.8. Formatowanie list wyliczeniowych i wypunktowań

Standardowo sposób formatowania list można parametryzować jak pokazano na rysunku 3.8. Jednak czasem trudno poradzić sobie z niektórymi rzeczami, jak np. znakami wypunktowania. Dlatego w szablonie wykorzystano pakiet `enumi`. Pozwala on na łatwe zarządzanie wyglądem list. W szablonie zastosowano następujące globalne ustawienia dla tego pakietu:

```
\usepackage{enumitem}
\setlist{noitemsep,topsep=4pt,parsep=0pt,partopsep=4pt,leftmargin=*}
\setenumerate{labelindent=0pt,itemindent=0pt,leftmargin=!,label=\arabic*.}
\setlistdepth{4}
\setlist[itemize,1]{label=$\bullet$}
\setlist[itemize,2]{label=\normalfont\bfseries\textendash}
\setlist[itemize,3]{label=$\ast$}
\setlist[itemize,4]{label=$\cdot$}
\renewlist{itemize}{itemize}{4}
```

W podrozdziale 4.2 pokazano przykład wykorzystania możliwości komend oferowanych w pakiecie `enumi`.

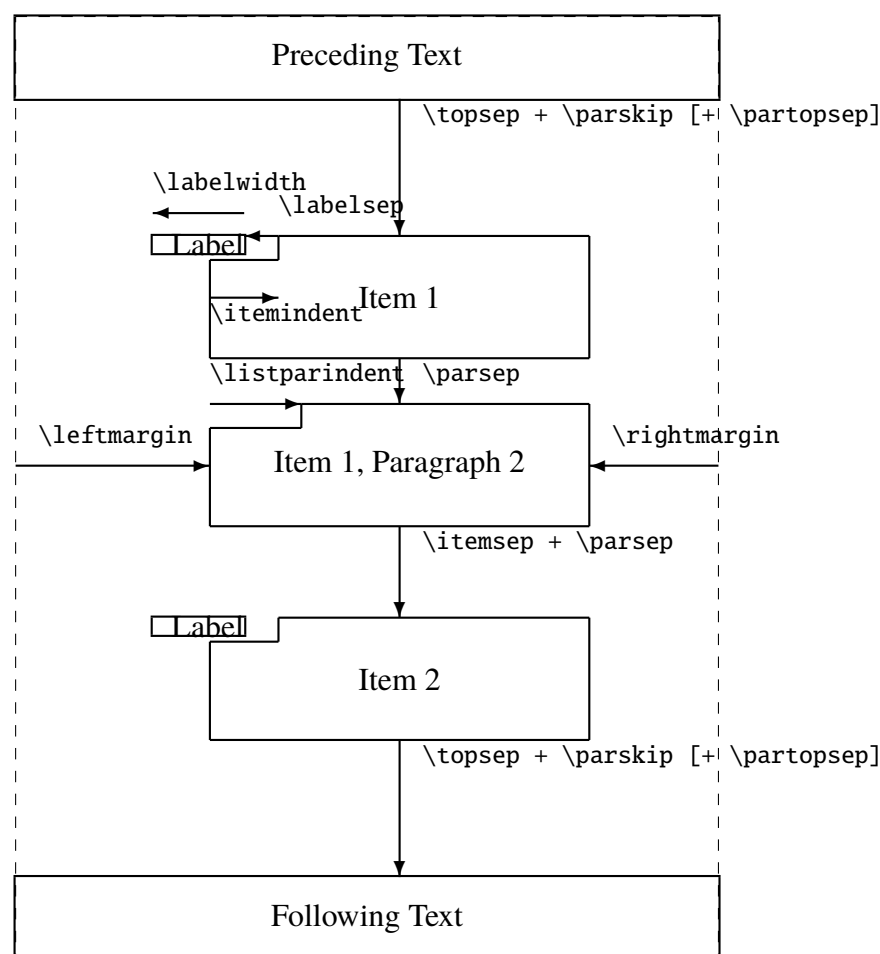


Rys. 3.7: Parametryzacja wyglądu spisu treści

3.9. Wzory matematyczne

Wzory matematyczne, jeśli mają być osobnymi formułami, powinny być wycentrowane, z numeracją umieszczoną na końcu linii i ujętą w okrągłe nawiasy (zobacz równanie (3.1)). Numery równań powinny zawierać numer rozdziału oraz kolejny numer równania w obrębie rozdziału (podobnie jak przy numerowaniu rysunków i tabel). Spełnienie tych warunków zapewnia otoczenie `equation`. Nie wszystkie formuły trzeba numerować (nienumerowane wzory można osiągnąć stosując otoczenie `\equation*`). Właściwie należy numerować tylko te, do których tworzy się jakieś odniesienia w tekście. Jeśli wzory umieszczane są w linijce tekstu, to można zastosować otoczenie matematyczne inline, jak w przykładzie $\int_0^{10\nu\sum i} x dx$ (wyprodukowanym komendą `\int_{0}^{10\nu\sum i}{x dx}`). Tylko że wtedy może dojść do rozszerzenia odstępów pomiędzy liniami tekstu (aby zmieścił się wzór).

$$\int_0^{10\nu\sum i} x dx \quad (3.1)$$



Rys. 3.8: Parametryzacja list wyliczeniowych i wypunktowań

Rozdział 4

Redakcja pracy

4.1. Układ pracy

Istnieje wiele sposobów organizacji poszczególnych rozdziałów pracy. W dokumentacji klasy memoir (<http://tug.ctan.org/tex-archive/macros/latex/contrib/memoir/memman.pdf>) można np. znaleźć zalecenia co do edytorskiej strony amerykańskich prac dyplomowych. Na stronie 363 można przeczytać, że część początkowa pracy może zawierać:

1. Title page
2. Approval page
3. Abstract
4. Dedication (optional)
5. Acknowledgements (optional)
6. Table of contents
7. List of tables (if there are any tables)
8. List of figures (if there are any figures)
9. Other lists (e.g., nomenclature, definitions, glossary of terms, etc.)
10. Preface (optional but must be less than ten pages)
11. Under special circumstances further sections may be allowed

A dalej kolejne wymagania, w tym informacje o zawartości części końcowej:

1. Notes (if you are using endnotes and grouping them at the end)
2. References (AKA 'Bibliography' or 'Works Cited')
3. Appendices
4. Biographical sketch (optional)

Prace dyplomowe powstające na Wydziale Elektroniki Politechniki Wrocławskiej standardowo redaguje się w następującym układzie:

Strona tytułowa
Streszczenie
Strona z dedykacją (opcjonalna)
Spis treści
Spis rysunków (opcjonalny)
Spis tabel (opcjonalny)
Spis listingów (opcjonalny)
Skróty (wykaz opcjonalny)
Abstrakt
1. Wstęp
1.1 Wprowadzenie
1.2 Cel i zakres pracy
1.3 Układ pracy
2. Kolejny rozdział
2.1 Sekcja
2.1.1 Podsekcja
Nienumerowana podpodsekcja
Paragraf
...
#. Podsumownie i wnioski
Literatura
A. Dodatek
A.1 Sekcja w dodatku
...
\$. Instrukcja wdrożeniowa
\$. Zawartość płyty CD/DVD
Indeks rzeczowy (opcjonalny)

Streszczenie – syntetyczny opis tego, o czym jest i co zawiera praca, zwykle jednostronicowy, po polsku i po angielsku (podczas wysyłania pracy do analizy antyplagiatowej należy skopiować tekst z tej sekcji do formularza w systemie ASAP).

Spis treści – powinien być generowany automatycznie, z podaniem tytułów i numerów stron. Typ czcionki oraz wielkość liter spisu treści powinny być takie same jak w niniejszym wzorcu.

Spis rysunków, Spis tabel – powinny być generowane automatycznie (podobnie jak Spis treści). Elementy te są opcjonalne (robienie osobnego spisu, w którym na przykład są tylko dwie pozycje specjalnie nie ma sensu).

Skróty – jeśli w pracy występują liczne skróty, to w tej części należy podać ich rozwinięcia.

Wstęp – pierwszy rozdział, w którym powinien znaleźć się opis dziedziny, w jakiej osadzona jest praca, oraz wyjaśnienie motywacji do podjęcia tematu (na to przeznaczona jest sekcja „Wprowadzenie”). W sekcji „Cel i zakres” powinien znaleźć się opis celu oraz zadań do wykonania, zaś w sekcji „Układ pracy” – opis zawartości kolejnych rozdziałów.

Podsumowanie – w rozdziale tym powinny być zamieszczone: podsumowanie uzyskanych efektów oraz wnioski końcowe wynikające z realizacji celu pracy dyplomowej.

Literatura – wykaz źródeł wykorzystanych w pracy (do każdego źródła musi istnieć odpowiednie cytowanie w tekście). Wykaz ten powinien być generowany automatycznie.

Dodatki – miejsce na zamieszczanie informacji dodatkowych, jak: Instrukcja wdrożeniowa, Instrukcja uruchomieniowa, Podręcznik użytkownika itp. Osobny dodatek powinien być przeznaczony na opis zawartości dołączonej płyty CD/DVD. Założono, że będzie to zawsze ostatni dodatek.

Indeks rzeczowy – miejsce na zamieszczenie kluczowych wyrazów, do których czytelnik będzie chciał sięgnąć. Indeks powinien być generowany automatycznie. Jego załączanie jest opcjonalne.

4.2. Styl

Zasady pisania pracy (przy okazji można tu zaobserwować efekt wyrównania wpisów występujących na liście wyliczeniowej uzależnione od długości etykiety):

1. Praca dyplomowa powinna być napisana w formie bezosobowej („w pracy pokazano ...”). Taki styl przyjęto na uczelniach w naszym kraju, choć w krajach anglosaskich preferuje się redagowanie treści w pierwszej osobie.
 2. W tekście pracy można odwołać się do myśli autora, ale nie w pierwszej osobie, tylko poprzez wyrażenia typu: „autor wykazał, że ...”.
 3. Odwołując się do rysunków i tabel należy używać zwrotów typu: „na rysunku pokazano ...”, „w tabeli zamieszczono ...” (tabela i rysunek to twory nieżywotne, więc „rysunek pokazuje” jest niepoprawnym zwrotem).
 4. Praca powinna być napisana językiem formalnym, bez wyrażen żargonowych („sejwowanie” i „downloadowanie”), nieformalnych czy zbyt ozdobnych („najznamienitszym przykładem tego niebywałego postępu ...”)
 5. Pisząc pracę należy dbać o poprawność stylistyczną wypowiedzi
 - trzeba pamiętać, do czego stosuje się „liczba”, a do czego „ilość”,
 - nie „szereg funkcji” tylko „wiele funkcji”,
 - redagowane zdania nie powinny być zbyt długie (lepiej podzielić zdanie wielokrotnie złożone na pojedyncze zdania),
 - itp.
 6. Zawartość rozdziałów powinna być dobrze wyważona. Nie wolno więc generować sekcji i podsekcji, które mają zbyt mało tekstu lub znacząco różnią się objętością. Zbyt krótkie podrozdziały można zaobserwować w przykładowym rozdziale 6.
 7. Niedopuszczalne jest pozostawienie w pracy błędów ortograficznych czy tzw. literówek – można je przecież znaleźć i skorygować automatycznie.
10005. Niedopuszczalne jest pozostawienie w pracy błędów ortograficznych czy tzw. literówek – można je przecież znaleźć i skorygować automatycznie.

Rozdział 5

Uwagi techniczne

5.1. Zasady redakcji

W pracy należy dbać o poprawność redakcyjną zgodnie z zaleceniami:

- nie zostawiać znaku spacji przed znakami interpunkcji („powiedziano , że ...” -> „powiedziano, że ...”),
- kropki po skrótach, które nie są jednocześnie kropkami kończącymi zdanie sklejać z kolejnym wyrazem znakiem tyldy, np. jak tutaj (np.~jak tutaj) lub wstawiać za nimi ukośnik, np. jak tutaj (np.\ jak tutaj)
- nie zapominać o dobrym sformatowaniu wyliczenia (należy zaczynać małymi literami lub dużymi oraz kończyć przecinkami, średnikami i kropkami – w zależności od kontekstu danego wyliczenia),
- nie zostawiać samotnych literek na końcach linii (można je „skleić” z wyrazem następnym stosując znaczek tyldy, jak w~przykładzie).
- nie zostawiać pojedynczych wierszy na końcu lub początku strony (należy kontrolować „sieroty” i „wdowy”),
- nie zostawiać odstępu pomiędzy tekstem a nawiasami czy znakami cudzysłowów (znaki te powinny przylegać do tekstu, który obejmują „jak w tym przykładzie”),
- wyrazy obcojęzyczne powinny być pisane czcionką pochyłą (preferowane `\emph{}`) wraz ze skrótem oznaczającym język, w szczególności ma to zastosowanie przy rozwijaniu skrótów, np. OGC (ang. *Open Geospatial Consortium*) (w kodzie latexowym wygląda to tak: np.~OGC (ang.~\emph{Open Geospatial Consortium})),
- każdy zastosowany skrót powinien zostać rozwinięty podczas pierwszego użycia, później może już występować bez rozwinięcia (skrót i jego rozwinięcie powinny trafić również do wykazu Skrótów, jeśli taki wykaz jest dołączany do dokumentu).
- nie wolno zostawiać zbyt dużo białej przestrzeni bez żadnego uzasadnienia.

Odnosząc się do ostatniej zasady, to przekłada się ona na gospodarowanie białą przestrzenią. Poniżej przedstawiono przykład złego użycia listy wyliczeniowej. Jeśli bowiem elementy na liście są reprezentowane przez krótki tekst, to wtedy powstaje biała plama. Widać to szczególnie przy długich listach.

Moje ulubione kolory to:

- biały,
- niebieski,
- czerwony.

Dużo lepiej w takim przypadku albo po prostu wyliczyć wartości po przecinkach:

Moje ulubione kolory to: biały, niebieski, czerwony.

albo wypisać listę w kilku kolumnach:

Moje ulubione kolory to:

- biały,
- niebieski,
- czerwony.

5.2. Rysunki

W niniejszym szablonie numeracja rysunków odbywa się automatycznie według następujących reguł: rysunki powinny mieć numerację ciągłą w obrębie danego rozdziału, sam zaś numer powinien składać się z dwóch liczb rozdzielonych kropką. Pierwsza liczbą ma być numer rozdziału, drugą – kolejny numer rysunku w rozdziale. Przykładowo: pierwszy rysunek w rozdziale 1 powinien mieć numer 1.1, drugi – numer 1.2 itd., pierwszy rysunek w rozdziale 2 powinien mieć numer 2.1, drugi – numer 1.2 itd. Wszystko to załatwia szablon.

Rysunki powinny być wyśrodkowane na stronie wraz z podpisem umieszczonym na dole. Podpisy nie powinny kończyć się kropką. Czcionka podpisu powinna być mniejsza od czcionki tekstu wiodącego o 1 lub 2 pkt (w szablonie jest to czcionka rozmiaru `small`). Ponadto należy zachowywać odpowiedni odstęp między rysunkiem, podpisem rysunku a tekstem rozdziału. Przykłady, jak to zrobić, zamieszczono w szablonie.

W przypadku korzystania z szablonu odstępy te regulowane są automatycznie. Podpis i grafika muszą stanowić jeden obiekt. Chodzi o to, że w edytorach tekstu typu Office podpis nie scala się z grafiką i czasem trafia na następną stronę, osieracając grafikę. Korzystającym z niniejszego szablonu i otoczenia `\figure` takie osierocenie nigdy się nie zdarzy.

Do każdego rysunku musi istnieć odwołanie w tekście (inaczej mówiąc: niedopuszczalne jest wstawienie do pracy rysunku bez opisu). Odwołania do rysunków powinny mieć postać: „Na rysunku 3.3 przedstawiono...” lub „... co ujęto na odpowiednim schemacie (rys. 1.7)”.

Jeśli odwołanie stanowi część zdania, to wtedy wyraz „rysunek” powinien pojawić się w całości. Jeśli zaś odwołanie jest ujęte w nawias (jak w przykładzie), wtedy należy zastosować skrót „rys.”. Jeśli do stworzenia obrazka wykorzystano jakieś źródła, to powinny one być zacytowane w podpisie tegoż rysunku.

Należy pamiętać o tym, że „rysunki” to twory nieżywotne. W związku z tym nie mogą „pokazywać”. Dlatego „rysunek 1.1 pokazuje ...” jest stylistycznie niepoprawne. Zamiast tego zwrotu trzeba użyć „na rysunku 1.1 pokazano ...”.

Rysunki można wstawiać do pracy używając polecenia `\includegraphics`. Zalecane jest, aby pliki z grafikami były umieszczane w katalogach odpowiadających numerom rozdziałów czy literom dodatków: `rys01`, `rysA` itd. Sposób wstawiania rysunków do pracy zademonstrowano na przykładzie rysunków 5.1 i 5.2.

Listing 5.1: Kod źródłowy przykładów wstawiania rysunków do pracy

```
\begin{figure}[ht]
  \centering
  \includegraphics[width=0.3\linewidth]{rys05/kanji-giri}
  \caption{Dwa znaki kanji - giri}
  \label{fig:kanji-giri}
\end{figure}

\begin{figure}[htb]
  \centering
  \begin{tabular}{@{}ll@{}}
    a) & b) \\
    \includegraphics[width=0.475\textwidth]{rys05/alfa1} &
  \end{tabular}
\end{figure}
```

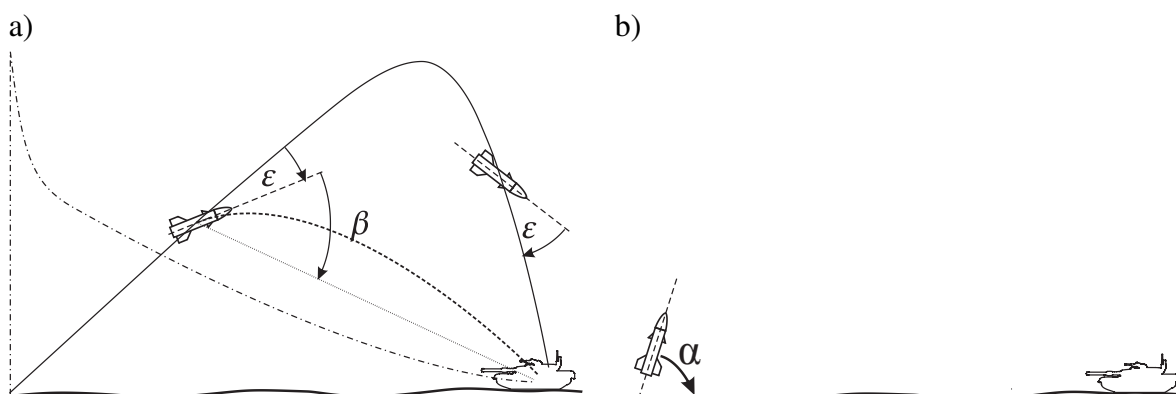
```

\includegraphics[width=0.475\textwidth]{rys05/beta1}
% jeśli obraki są różnej wysokości, można je wyrównać do góry
% ↪ stosując vtop jak niżej
% \vtop{\vskip-2ex\hbox{\includegraphics[width=0.475\textwidth]{
% ↪ rys05/beta1}}}} &
% \vtop{\vskip-2ex\hbox{\includegraphics[width=0.475\textwidth]{
% ↪ rys05/alfa1}}}}
\end{tabular}
a) trzy podejścia, b) podejście praktyczne}
\label{fig:alfabeta}
\end{figure}

```

義理

Rys. 5.1: Dwa znaki kanji – giri



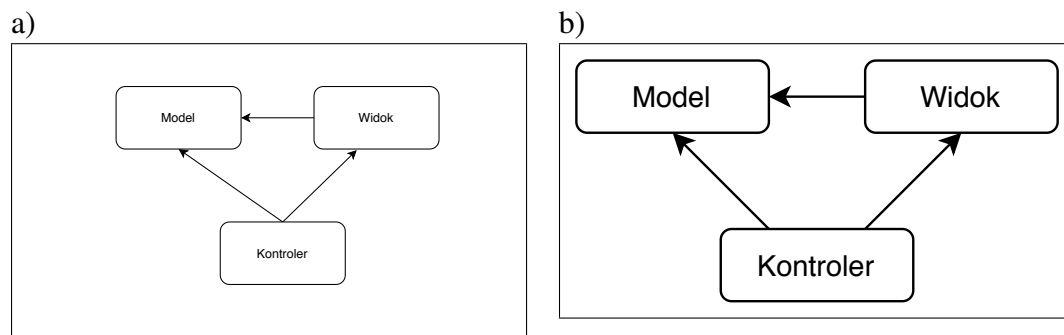
Rys. 5.2: Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne

Grafiki wektorowe powinny być dostarczone w plikach o formacie pdf. Rozmiar strony w pliku pdf powinien być troszeczkę większy niż zamieszczona na nim grafika (proszę spojrzeć na przykłady grafik wykorzystanych w niniejszym szablonie). Chodzi o to, aby na rysunku nie pojawiała się niepotrzebna biała przestrzeń (rozmiar płótna ma odpowiadać rozmiarowi grafiki bez żadnych marginesów, elementy grafiki powinny być ciasno ułożone). Grafiki rastrowe (głównie zrzuty z ekranu bądź zdjęcia) powinny być dostarczane w plikach o formacie png z kompresją bezstratną. Zastosowanie kompresji stratnej, jak jpg, wprowadza niepotrzebne artefakty. Podobnie jak w przypadku grafik wektorowych, grafiki rastrowe nie powinny mieć białych marginesów.

Nieźłą ścieżką zapewniającą wygenerowanie ładnej grafiki wektorowej, w szczególności diagramów, jest:

- w diagrams.net (dawniej draw.io): narysowanie diagramu i wyeksportowanie do pdf
- w inkscape: zaimportowanie pdf, rozdzielenie grupy, wykasowanie niepotrzebnych elementów (tła), zaznaczenie wszystkiego, przycięcie strony do zaznaczonych (Ctrl-Shift-R), zapisanie jako pdf.

Na rysunku 5.3 pokazano przykład dobrze i źle (od strony technicznej) narysowanego diagramu. Na rysunku celowo pokazano ramki, by było widać marginesy. Normalnie ramek tych nie należy stosować.



Rys. 5.3: Przykład diagramu: a) złego, b) w miarę dobrego

Na rysunkach nie powinno stosować się 100% czarnego wypełnienia, bo robią się plamy przebijające się przez kartkę. Zamiast tego wypełnienie powinno być ok. 90% czerni.

Czcionka na rysunkach nie może być większa od czcionki wiodącej tekstu (jedyne wyjątek to np. jakieś nagłówki). Należy stosować czcionkę kroju Arial, Helvetica bądź tego samego kroju co czcionka dokumentu (`texgyre-termes`).

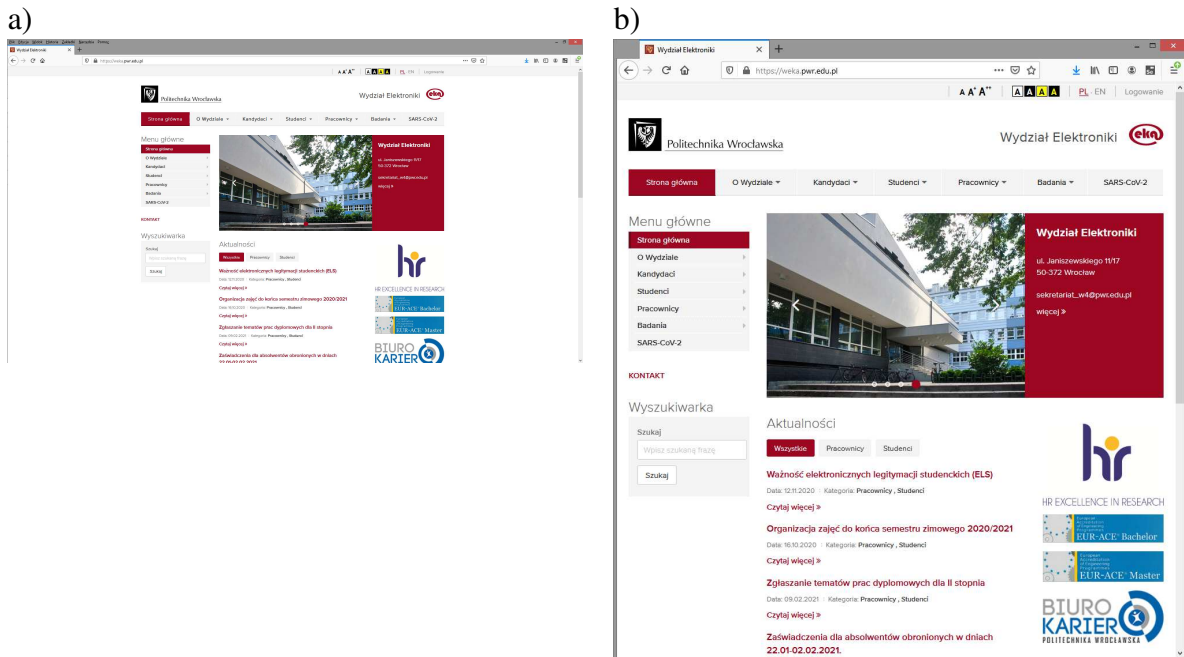
Jeśli na jednym rysunku pojawić się ma kilka grafik, to zamiast stosować `subfigure` lub inne otoczenia należy wstawić grafiki w tabelę, opisać ją indeksami a) i b), a potem odnieść się do tego w podpisie (rys. 5.2). Czasem pomaga w pozycjonowaniu rysunków użycie komendy:

```
\vtop{\vskip-2ex\hbox{\includegraphics[width=0.475\textwidth]{nazwa}}}
```

Na rysunkach nie wolno nadużywać kolorów oraz ozdóbników (wiele narzędzi do tworzenia diagramów dostarcza grafikę z cieniowaniem, gradacją kolorów itp. co niekoniecznie przekłada się na czytelność rysunku). Jeśli rysunki są kolorowe, to kolory te powinny być rozróżnialne po konwersji do poziomów szarości (chodzi o to, aby na wydrukach wykonanych na drukarkach monochromatycznych można było dostrzec różnice).

Podczas robienia zrzutów z ekranu należy zadbać o to, by taki zrzut był czytelny po wydrukowaniu. Czyli aby pojawiające się literki były wystarczająco duże, a przestrzeń bez treści – relatywnie małe. Przystępując do robienia zrzutu trzeba odpowiednio wyskalować elementy na ekranie. Na przykład robiąc zrzut z przeglądarki FF najpierw należy wcisnąć CTR-0 (domyślne skalowanie), potem CTR-- (zmniejszenie skali o stopień). Potem dobrze jest zawęzić okno przeglądarki tak, by interesująca treść wypełniła je w całości. Jeśli na obserwowanej stronie jest zbyt dużo pustych obszarów, to należy je jakoś zawęzić (sterując wielkością okna przeglądarki lub aktywnymi elementami interfejsu użytkownika). Zrzut bowiem wcale nie musi być odwzwierciedleniem 1:1 domyślnego układu obserwowanych elementów. Ważne jest, by na zrzucie pokazać interesujący, opisywany fragment i żeby ten fragment był czytelny. Nie trzeba też zawsze robić zrzutów w układzie 16:9 (lub innym panoramicznym). Czasem lepiej jest zrobić zrzuty okien niemal kwadratowych, bo lepiej się układają w wynikowym dokumencie. Poza tym można je przeskalować (powiększyć) by zajmowały całą szerokość strony, a wtedy czcionka na wydrukach będzie większa. Takie skalowanie dla zrzutów panoramicznych zwykle się nie udaje. Na rysunku 5.4 pokazano przykłady dobrze i źle zrobionych zrzutów (w celu oszczędzenia miejsca zrzuty umieszczono obok siebie).

Czasem problemem jest tworzenie zrzutów z ekranu, gdy występują na nim dane wrażliwe. Istnieją dwa sposoby na radzenie sobie z tym problemem. Pierwszy polega na zastąpieniu w systemie danych rzeczywistych danymi testowymi – wygenerowanymi tylko do celów prezentacji. Zrzut robi się wtedy na bazie danych testowych. Drugi polega na wykonaniu zrzutu z ekranu, na którym pokazano dane rzeczywiste, i następnie zamianie tych danych już w pliku graficznym za pomocą odpowiedniego edytora (np. `gimp`). Czyli oryginalny zrzut z ekranu należy otworzyć w edytorze, a potem nadpisać oryginalny tekst własnym tekstem. Konieczne jest wtedy dobranie odpowiednich czcionek aby nie było widać wprowadzonych zmian.



Rys. 5.4: Przykłady zrzutów z ekranu: a) zły (nieczytelny, zrobiony przy zbyt szerokim oknie, z niepotrzebnymi marginesami, niepotrzebnym paskiem menu), b) w miarę dobry (w miarę czytelny, zrobiony przy zawężonym oknie, byłoby wskazane jeszcze usunięcie z niego beleczki z faviconem (jeśli nic nie wnosi) oraz przycięcie od dołu (jeśli treści tam pokazywane nie są istotne))

Uwaga: takie manipulowanie zrzutami jest usprawiedliwione jedynie w przypadku konieczności ochrony danych wrażliwych czy też lepszego pokazania wybranych elementów. Nie może to prowadzić generowania fałszywych rezultatów!!!

5.3. Wstawianie kodu źródłowego

Kod źródłowy można wstawiać jako blok tekstu pisany czcionką maszynową. Używa się do tego otoczenie `\lstlisting`. W atrybutach otoczenia można zdefiniować tekst podpisu wstawianego wraz z numerem nad blokiem, etykietę do tworzenia odwołań, sposób formatowania i inne ustawienia. Zaleca się stosowanie w tym otoczeniu następujących parametrów:

```
\begin{lstlisting}[label=list:req1,caption=Initial HTTP Request,
basicstyle=\footnotesize\ttfamily]
```

Szczególnie przydatne podczas wstawiania większej ilości kodu źródłowego jest zastosowanie parametru `basicstyle=\footnotesize\ttfamily`. Dzięki niemu zmniejsza się czcionka, a przez to na stronie można zmieścić dłuższe linijki kodu. Użycie tak zdefiniowanego parametru nie jest jednak sztywnym zaleceniem. Wielkość czcionki można dobierać do potrzeb.

Listing 5.2: Initial HTTP Request

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Można też sformatować kod bez stosowania numerowanego podpisu (wtedy nie zamieszcza się `caption` na liście atrybutów).

```
GET /script/Articles/Latest.aspx HTTP/1.1
Host: www.codeproject.com
Connection: keep-alive
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml
User-Agent: Mozilla/5.0 ...
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US...
Accept-Charset: windows-1251,utf-8...
```

Istnieje możliwość wstawiania kodu źródłowego w bieżącej linii tekstu. Można to zrobić na kilka sposobów:

- korzystając z polecenia `\texttt` ustawiającego czcionkę maszynową, jak w przykładzie tutaj (efekt zastosowania komendy `\texttt{tutaj}`). Problemem jednak mogą okazać się znaki podkreślenia i inne znaki kontrolne.
- korzystając z otoczenia `\verb` zapewniającego wypisanie kodu czcionką maszynową jak w przykładzie tutaj (efekt zastosowania komendy `\verb|tutaj|`). Problemem jest to, że polecenie `\verb` nie potrafi łamać dłuższego tekstu.
- korzystając z polecenia `\lstin` umożliwiającego wypisanie kodu czcionką ustawianą w opcjach jak w przykładzie tutaj (efekt komendy `\lstset{basicstyle=\ttfamily}\lstinline{tutaj}`) lub tutaj (efekt komendy `\lstinline[basicstyle=\ttfamily]=tutaj=`).

Poniżej zamieszczono przykłady kodów źródłowych z podświetleniem składni.

Listing 5.3: Opis 1

```
package pl.mrbarzoit.backend;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class BackendApplication {

    public static void main(String[] args) {
        SpringApplication.run(BackendApplication.class, args);
    }

}
```

Jeśli w kodzie źródłowym jest jakiś nieistotny fragment względem omawianego problemu, to można go wykopkować (patrz listing 5.4).

Listing 5.4: Opis 2

```
// Karma configuration file, see link for more information
// https://karma-runner.github.io/1.0/config/configuration-file.html

module.exports = function (config) {
    config.set({
        basePath: '',
        frameworks: ['jasmine', '@angular-devkit/build-angular'],
        plugins: [
            require('karma-jasmine'),
            require('karma-chrome-launcher'),
            require('karma-jasmine-html-reporter'),
            require('karma-coverage-istanbul-reporter'),
```

```

    require('@angular-devkit/build-angular/plugins/karma')
  ],
  ... // opuszczony kod
  autoWatch: true,
  browsers: ['Chrome'],
  singleRun: false,
  restartOnFileChange: true
});
};

```

Jeśli kod jest wąski, to można taki kod wrzucić w dwie kolumny. Przykład zamieszczono na listingu 5.5.

Listing 5.5: Kontrakt na model wejściowy endpointu `/api/v1/chess/board/move`.

```

1  {
2    "lastPosition": {
3      "fenDescription": "string"
4    },
5    "image": {
6      ...
7    },
8    "positions": {
9      "chessboardCorners": [
10     ...
11     ],
12     "tilesCornerPoints": [
13       ...
14     ]
15   },
16   "referenceColors": {
17     ...
18   },
19 }

```

5.4. Wykaz literatury oraz cytowania

Cytowania powinny być zamieszczane w tekście z użyciem komendy `\cite{}`. Jej argumentem powinien być klucz cytowanej pozycji (lub lista kluczy rozdzielonych przecinkiem bez spacji, jeśli takich pozycji w danym miejscu cytuje się więcej) jaki jest używany w bazie danych bibliograficznych (plik dokumentacja.bib). Po kompilacji bibtex i pdflatex w tekście pojawia się właściwy odsyłacz do pozycji w wykazie literatury (ujęty w kwadratowe nawiasy – zgodnie z tym, co definiuje styl `plabrv.bst`), zaś w samym wykazie (rozdział Literatura) – zacytowana pozycja. Przykładem cytowania jest: „dobrze to opisano w pracach [2, 1]” (gdzie zastosowano komendę `\cite{JS07,SQL2}`).

Co do zawartości rekordów bibliograficznych - style bibtexowe potrafią „skracać” imiona (czyli wstawiać, jeśli taka wola, inicjały zamiast pełnych imion). Niemniej dobrze jest od razu przyjąć jakąś konwencję. Proponuje się, aby w rekordach od razu wstawiane były inicjały zamiast pełnych imion.

Niekiedy tytuły prac zawierają wyrazy z dużymi i małymi literami. Takie tytuły należy brać w podwójne nawiasy klamrowe, aby bibtex nie zamienił ich na postać, w której poza pierwszą literą pozostałe są małe.

Jeśli jakiś cytowany zasób pochodzi z Internetu, to jego rekord w pliku bib powinien wyglądać jak niżej.

```

@INPROCEEDINGS{SQL2,
  title={A MySQL-based data archiver: preliminary results},
  author={Bickley, M. and Slominski, Ch.},
  booktitle = {{Proceedings of ICALEPCS07}},
  month = oct,
  day = {15--19},
  year={2007},
  note={\url{http://www.osti.gov/scitech/servlets/purl/922267}
    [dostęp dnia 20 czerwca 2015]}
}

```

A to inny przykład rekordu danych bibliograficznych:

```
@TechReport{JS07,
  author = {Jędrzejczyk, J. and Śródka, B.},
  title   = {Segmentacja obrazów metodą drzew decyzyjnych},
  year    = {2007},
  institution = {Politechnika Wrocławska, Wydział Elektroniki}
}
```

5.5. Indeks rzeczowy

Generowanie indeksu po trosze wygląda jak generowanie wykazu literatury – wymaga kilku kroków. Podczas pierwszej kompilacji `pdflatex` generowany jest plik z rozszerzeniem `*.idx` (zawierający „surowy indeks”). Następnie, bazując na tym pliku, generowany jest plik z rozszerzeniem `*.ind` zawierający sformatowane dane. Ten krok wymaga uruchomienia odpowiedniego narzędzia oraz zastosowania pliku z definicją stylu `Dyplom.ist`. W kroku ostatnim dokonuje się kolejnej kompilacji `pdflatex` (dzięki niej w wynikowym dokumencie pojawi się Indeks rzeczowy). Domyślnie Indeks rzeczowy zostanie sformatowany w układzie dwukolumnowym.

Oczywiście aby to wszystko zadziało w kodzie szablonu należy umieścić odpowiednie komendy definiujące elementy indeksu rzeczowego (`\index`) oraz wstawiające sformatowany Indeks rzeczowy do dokumentu wynikowego (`\printindex`). Więcej informacji o tworzeniu indeksu rzeczowego można znaleźć na stronie <https://en.wikibooks.org/wiki/LaTeX/Indexing>. Poniżej przedstawiono przykłady komend użytych w szablonie do zdefiniowania elementów indeksu rzeczowego:

- `\index{linia komend}` – pozycji główna.
- `\index{generowanie!-- indeksu}` – podpozycja.

Generowanie pliku `*.ind` można inicjować na kilka sposobów:

- poprzez wydanie odpowiedniego polecenia bezpośrednio w linii komend
`makeindex Dyplom.idx -t Dyplom.ilg -o Dyplom.ind -s Dyplom.ist`
- poprzez odpalenie odpowiedniego narzędzia środowiska. Na przykład w `TeXnicCenter` definiuje się tzw. `output profiles`:

```
makeindex "%tm.idx" -t "%tm.ilg" -o "%tm.ind" -s "%tm.ist"
```

a samo generowanie pliku `*.ind` zapewni wybranie pozycji menu `Build/Makeindex`.

- korzystając z odpowiednio sparametryzowanych pakietów i komend wewnątrz kompilowanego dokumentu (czyli od razu przy okazji jego kompilacji).

```
\DisemulatePackage{imakeidx}
\usepackage[noautomatic]{imakeidx}
% jeśli chcemy, by indeks by generowany automatycznie programem makeindex
↪ :
%\usepackage[makeindex]{imakeidx}
% a tak ponoć można przekazać opcje do programu generującego indeks:
%\makeindex[options=-s podrecznik -L polish -M lang/polish/utf8]
%\makeindex[options=-s podrecznik]
\makeindex
```

Niestety, `makeindex` jest narzędziem, które umieszcza część pozycji w grupie `Symbols`, a nie w grupach związanych z literkami alfabetu. W związku z czym indeksowany element zaczynający się od polskiej literki trafia do grupy `Symbols`, jak np. `\index{Światło}`. Jeśli chce się zamieszczać w indeksie symbole matematyczne, to dobrze jest to robić jak w następującym przykładzie: `\index{$asterisk@$ \ast$}` czy też `\index{c@$ \mathcal{C}$}`,

tj. dostarczając przy okazji klucz do sortowania. Lepiej w tym względzie radzą sobie inne narzędzia, jak `texindy` lub `xindy` dostępne pod linuxem. Korzystając z nich uzyskuje się grupy polskich literek w indeksie rzeczowym (hasła zaczynające się od polskich literek już nie trafiają do grupy Symbols). Przykład polecenia wydanego z linii komend, w którym wykorzystano `texindy` zamieszczono poniżej (zakładamy kodowanie plików w UTF8, można dla niniejszego szablonu zmienić na cp1250):

```
texindy -L polish -M lang/polish/utf8 Dyplom.idx
```

To polecenie wygeneruje `Dyplom.ind` o zawartości:

```
\begin{theindex}
  \providecommand*\lettergroupDefault[1]{}
  \providecommand*\lettergroup[1]{%
    \par\textbf{#1}\par
    \nopagebreak
  }

  \lettergroup{G}
  \item generowanie
    \subitem -- indeksu, 27
    \subitem -- wykazu literatury, 27

  \indexspace

  \lettergroup{L}
  \item linia komend, 27

  \indexspace

  \lettergroup{Ś}
  \item \'Swiat\IeC {\l }o, 28

\end{theindex}
```

Aby mieć większą kontrolę automatyczne generowanie indeksu zostało w niniejszym szablonie wyłączone (indeks trzeba wygenerować samemu, wydając polecenie `makeindex` lub zalecane `texindy`).

5.6. Inne uwagi

Dobrym sposobem na kontrolę błędów występujących podczas kompilacji jest wstawianie linijki `\end{document}` w wybranym miejscu dokumentu. Jest to szczególnie przydatne w przypadkach, gdy błędy te są trudne do zidentyfikowania (gdy wygenerowane przez kompilator numery linii z błędami nie są tymi, w których błędy występują). Wystarczy wtedy przestawić wspomnianą linijkę do kolejnych miejsc, aż znajduję to miejsce, gdzie występuje problem.

Aby osiągnąć apostrofy maszynowe (czyli takie złożone z samych kresek) należy użyć polecenia `"{}jak tutaj{}"` (podwójny apostrof i podwójny apostrof z na wszelki wypadek umieszczonymi nawiasami klamrowymi, nawiasy są potrzebne z tej racji, iż podwójny apostrof przed niektórymi literkami zamienia je na literki z akcentami). W efekcie otrzymamy "jak tutaj". Jeśli natomiast apostrofy mają być drukarskie (czyli złożone z kropek i kresek), to należy użyć polecenia `„,jak tutaj”` (dwa pojedyncze przecinki i dwa pojedyncze apostrofy). W efekcie otrzymamy „jak tutaj”. Można też użyć znaków apostrofów odpowiednio zakodowanych „jak tutaj”, tylko że czasem trudno pisać takie apostrofy w środowiskach kompilacji projektów latexowych.

Oto sposoby ustawienia odstępów między liniami:

- używając komendy `\linespread{...}` (akceptowalne), przy czym atrybutem tej metody jest współczynnik zależny od wielkości czcionki. Dla czcionki wiodącej 12pt odstęp półtora linii osiągnie się komendą `\linespread{1.241}`. Dla innych czcionek wiodących wartości tego parametru są jak w poniższym zestawieniu.

```
10pt 1.25 dla \onehalfspacing
      1.667 for \doublespacing,
      ponieważ „basic ratio” = 1.2
      (\normalfont posiada \baselineskip rozmiaru 12pt)
11pt 1.213 dla \onehalfspacing oraz 1.618 dla \doublespacing,
      ponieważ „basic ratio” = 1.236
      (\normalfont posiada \baselineskip rozmiaru 13.6pt)
12pt 1.241 dla \onehalfspacing oraz 1.655 dla \doublespacing,
      ponieważsince „basic ratio” is 1.208
      (\normalfont has a \baselineskip of 14.5pt)
```

Kłopot w tym, że raz ustawiony odstęp będzie obowiązywał do wszystkich czcionek (nie działa tu żaden mechanizm zmiany współczynnika w zależności od wielkości czcionki akapitu).

- używając pakietu `setspace` (niezalecane). Ponieważ klasa `memoir` emuluje pakiet `setspace`, w preambule dokumentu należałoby umieścić:

```
\DisemulatePackage{setspace}
\usepackage{setspace}
```

a potem można już sterować odstęp komendami:

```
\singlespacing
\onehalfspacing
\doubelspacing
```

Ten sposób pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

- korzystając bezpośrednio z komend dostarczonych w klasie `memoir` (zalecane):

```
\SingleSpacing
\OnehalfSpacing
\DoubleSpacing
```

Ten sposób również pozwala na korzystanie z mechanizmu automatycznej zmiany odległości linii w zależności od wielkości czcionki danego akapitu.

Na koniec jeszcze uwaga o rozmiarze pliku wynikowego. Otóż `pdflatex` generuje pliki pdf, które zazwyczaj mogłyby być nieco lepiej skompresowane. Do lepszego skompresowania tych plików można użyć programu `ghostscript`. Wystarczy w tym celu wydać komendę (pod windowsami):

```
gswin64 -sDEVICE=pdfwrite -dCompatibilityLevel=1.4 -dNOPAUSE -dQUIET \
-dSAFER -dBATC -sOutputFile=Dyplom-compressed.pdf Dyplom.pdf
```

W poleceniu tym można również wstawić opcję `-dPDFSETTINGS=/prepress` (zapewniającą uzyskanie wysokiej jakości, zachowanie kolorów, uzyskanie obrazków w rozdzielczości 300 dpi). Ze względów licencyjnych `ghostscript` używa domyślnie algorytmów z kompresją stratną. Przy kompresji może więc dojść do utraty jakości bitmap.

Rozdział 6

Podsumowanie

Podsumowanie jest miejscem, w którym należy zamieścić syntetyczny opis tego, o czym jest dokument. W szczególności w pracach dyplomowych w podsumowaniu powinno znaleźć się jawnie podane stwierdzenie dotyczące stopnia realizacji celu. Czyli powinny pojawić się w niej akapity ze zdaniami typu: „Podczas realizacji pracy udało się zrealizować wszystkie postawione cele”. Ponadto powinna pojawić się dyskusja na temat napotkanych przeszkód i sposobów ich pokonania, perspektyw dalszego rozwoju, możliwych zastosowań wyników pracy itp.

6.1. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Nam id nulla a adipiscing tortor, dictum ut, lobortis urna. Donec non dui. Cras tempus orci ipsum, molestie quis, lacinia varius nunc, rhoncus purus, consectetur congue risus.

6.1.1. Sekcja poziomu 2

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Sekcja poziomu 3

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Paragraf 4 Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

6.2. Sekcja poziomu 1

Lorem ipsum dolor sit amet eleifend et, congue arcu. Morbi tellus sit amet, massa. Vivamus est id risus. Sed sit amet, libero. Aenean ac ipsum. Mauris vel lectus.

Literatura

- [1] M. Bickley, C. Slominski. A MySQL-based data archiver: preliminary results. *Proceedings of ICALEPCS07*, Paz. 2007. <http://www.osti.gov/scitech/servlets/purl/922267> [dostęp dnia 20 czerwca 2015].
- [2] J. Jędrzejczyk, B. Śródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

Dodatek A

Instrukcja wdrożeniowa

Jeśli praca skończyła się wykonaniem jakiegoś oprogramowania, to w dodatku powinna pojawić się instrukcja wdrożeniowa (o tym jak skompilować/zainstalować to oprogramowanie). Przydałoby się również krótkie how to (jak uruchomić system i coś w nim zrobić – zademonstrowane na jakimś najprostszym przypadku użycia). Można z tego zrobić osobny dodatek,

Dodatek B

Opis załączonej płyty CD/DVD

Tutaj jest miejsce na zamieszczenie opisu zawartości załączonej płyty. Należy wymienić, co zawiera.