

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Automatyka i Robotyka (AIR)  
SPECJALNOŚĆ: Przemysł 4.0 (ARP)

PRACA DYPLOMOWA  
INŻYNIERSKA

Serwis internetowy wspomagający planowanie  
wyprowadzania psów na spacer

A website supporting the planning of taking  
dogs for a walk

AUTOR:

Amadeusz Janiszyn

PROWADZĄCY PRACĘ:

Dr inż. Mariusz Uchroński,  
Katedra Automatyki, Mechatroniki  
i Systemów Sterowania

OCENA PRACY:

Opracował: Amadeusz Janiszyn <249013@student.pwr.edu.pl>  
Data: Grudzień 2021



Szablon jest udostępniany na licencji Creative Commons: *Uznanie autorstwa – Użycie niekomercyjne – Na tych samych warunkach, 3.0 Polska*, Wrocław 2016.

Oznacza to, że wszystkie zawarte nim 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/>.

# Spis treści

<b>1. Wstęp</b>	<b>5</b>
1.1. Cel oraz zakres pracy	5
1.2. Główne założenia projektowe	6
1.3. Układ pracy	6
<b>2. Część teoretyczna</b>	<b>7</b>
2.1. Wprowadzenie do projektu	7
2.2. Proces projektowania i implementacji	7
2.3. Wymagania funkcjonalne oraz нефункционалне	7
2.3.1. Wymagania funkcjonalne	7
2.3.2. Wymagania нефункционалне	8
2.4. Diagram oraz przypadki użycia	8
2.4.1. Przypadki użycia	8
2.5. Widoki, mockupy	13
2.6. Backend	13
2.7. Frontend	14
<b>3. Zalecenia dotyczące formatowania</b>	<b>16</b>
3.1. Rozmiar i układ treści na stronach dokumentu	16
3.2. Strona tytułowa	18
3.3. Krój i wielkość czcionek	19
3.4. Formatowanie bloków tekstu	20
3.5. Opisy tabel i rysunków	22
3.6. Przypisy dolne	23
3.7. Formatowanie spisu treści	23
3.8. Formatowanie list wyliczeniowych i wypunktowań	23
3.9. Wzory matematyczne	25
<b>4. Redakcja pracy</b>	<b>26</b>
4.1. Układ pracy	26
4.2. Styl	27
<b>5. Uwagi techniczne</b>	<b>28</b>
5.1. Rysunki	28
5.2. Wstawianie kodu źródłowego	30
5.3. Wykaz literatury oraz cytowania	31
5.4. Indeks rzeczowy	32
5.5. Inne uwagi	33
<b>6. Podsumowanie</b>	<b>35</b>

---

6.1. Sekcja poziomu 1 . . . . .	35
6.1.1. Sekcja poziomu 2 . . . . .	35
6.2. Sekcja poziomu 1 . . . . .	35
<b>Literatura . . . . .</b>	<b>38</b>
<b>A. Tytuł dodatku . . . . .</b>	<b>39</b>
<b>B. Opis załączonej płyty CD/DVD . . . . .</b>	<b>40</b>
<b>Indeks rzeczowy . . . . .</b>	<b>41</b>

# Rozdział 1

## Wstęp

Globalny dostęp do Internetu spopularyzował rozpowszechnianie się aplikacji oraz serwisów internetowych. Obecnie coraz więcej przedsiębiorstw decyduje się na implementację swoich rozwiązań w takiej postaci. Zaletą takich rozwiązań są minimalne wymagania sprzętowe użytkownika - aplikacja stoi na zewnętrznym serwerze, który komunikuje się z klientem za pośrednictwem protokołu http, zatem od użytkownika wymagane jest urządzenie z dostępem do internetu.

Zastosowanie aplikacji internetowych pozwala zatem dotrzeć z potencjalnym produktem do większej liczby konsumentów bez konieczności instalowania dodatkowego oprogramowania. Wystarczą urządzenia, które użytkujemy na co dzień – od komputerów aż po smartfony.

Obecnie dużą popularnością cieszy się podejście do tworzenia tzw. Single-Page Application. Czyli aplikacji, których głównym założeniem jest wysyłanie do użytkownika jednego pliku HTML, który na bieżąco podmienia swoją zawartość z poziomu przeglądarki. W trakcie korzystania z programów, strony internetowe nie są przeładowywane, co powoduje szybsze działanie aplikacji oraz pozytywnie wpływa na doświadczenia użytkowników. W przypadku starego podejścia, tzn Multi-Page Application, pliki były generowane na bieżąco po stronie serwera i wysyłane do przeglądarki co znacząco wydłużało czas ładowania strony.

### 1.1. Cel oraz zakres pracy

Celem pracy jest zaprojektowanie i zaimplementowanie serwisu internetowego w oparciu o nowoczesne technik programowania, którego zadaniem jest wspomaganie planowania wyprowadzania psów na spacer. Aplikacja skonstruowana jest w oparciu o dwie główne grupy docelowych użytkowników – właścicieli oraz opiekunów.

Zakres pracy obejmuje stworzenie aplikacji przy użyciu języka Java do stworzenia serwera oraz aplikacji dostępowej stworzonej przy pomocy frameworka Angular. Serwis pobiera i zapisuje dane przy użyciu MongoDB – nowoczesnej oraz nierelacyjnej bazy danych, która w ciągu ostatnich lat zdobywa coraz większą popularność. Wykorzystano również szereg narzędzi wspomagających i usprawniających tworzenie aplikacji.

## 1.2. Główne założenia projektowe

- Właściciele dostają szereg funkcjonalności, pozwalających na stworzenie profilu zwierzaka oraz zaplanowanie spacerów z deklaracją godziny oraz miejsca spaceru. Możliwe jest również wystawianie opinii po spacerze w celu weryfikacji kompetencji opiekuna;
- Opiekunowie otrzymują możliwość wyszukiwania spacerów, która bazuje na lokalizacji użytkownika. Aplikacja udostępnia tygodniowy oraz miesięczny wygląd planera ze spacerami, na które użytkownik się zapisał. Opiekunowie również mogą dodawać informacje o odbytym spacerze;
- Administracja dostaje możliwość podglądu bazy użytkowników, zwierząt oraz spacerów, jak i monitorowania samej aplikacji oraz błędów, które zgłosili inni użytkownicy;

## 1.3. Układ pracy

Praca została podzielona na niżej wymienione rozdziały:

- Część teoretyczna, opisuje dokładnie wykorzystane technologie, techniki oraz programy, które zostały wykorzystane przy implementacji pracy;
- Część implementacyjna, skupia się na szczegółach dotyczących implementacji aplikacji oraz zawiera opis funkcjonalny;
- Podsumowanie, streszcza proces tworzenia aplikacji, oraz prezentuje dalsze możliwości rozwoju aplikacji w przyszłości;

# Rozdział 2

## Część teoretyczna

### 2.1. Wprowadzenie do projektu

Genezą powstania serwisu internetowego jest fakt, że istnieje duża grupa osób z psami, które z różnych przyczyn nie mają czasu na regularne spacery ze swoim pupilem. Dodatkową funkcjonalnością oferowaną przez aplikację jest możliwość wyprowadzania potrzebujących psów na spacer jako opiekun. Jest to opcja przeznaczona dla osób, które lubią zajmować się zwierzętami i mają więcej wolnego czasu, niż opiekunowie.

### 2.2. Proces projektowania i implementacji

Proces tworzenia aplikacji obejmował kilka etapów:

- Opracowanie wymagań funkcjonalnych oraz нефункциональных;
- Opracowanie widoków aplikacji;
- Opracowanie warstwy biznesowej po stronie backendu;
- Opracowanie warstwy dostępowej po stronie frontendu;

Organizacja pracy nad projektem odgrywa kluczowe znaczenie na każdym etapie implementacyjnym. Każde stadium projektu było zarządzane przy pomocy aplikacji Trello. Jest to darmowa aplikacja, pozwalająca na skuteczne kierowanie projektem przez jedną lub wiele osób, współpracujących ze sobą. Organizacja pracy polegała na tworzeniu tablic do poszczególnych etapów. Tablice były tworzone na podstawie tych używanych w metodologii Kanban. Następnie na początku każdego etapu były planowane zadania do zrobienia i wpisywane do aplikacji. Miało to na celu większą organizację oraz kontrolę nad projektem. Wczesne zaznajomienie się ze zwinnymi metodykami zarządzania produkcją jest obecnie cenione na rynku pracy, ponieważ wiele firm z sektora IT używa ich na codzień.

### 2.3. Wymagania funkcjonalne oraz нефункциональные

W pierwszym etapie projektowania serwisu został stworzony spis wymagań funkcjonalnych oraz нефункциональных. Miało to na celu pogrupowanie zadań na odpowiednie podetapy. Pozwoliło skupić się na kluczowych, dla danej fazy, funkcjach.

#### 2.3.1. Wymagania funkcjonalne

- Rejestracja nowych użytkowników z podziałem na role;
- Logowanie użytkowników do serwisu;

- Możliwość zmiany danych użytkownika – zdjęcie profilowe, dane osobowe, hasło;
- Możliwość dodania profilu swojego psa;
- Możliwość stworzenia spaceru;
- Ocena opiekunów po zakończonym spacerze;
- Przeglądanie listy dostępnych spacerów;
- Przeglądanie listy nadchodzących spacerów;
- Zapisywanie na spacer;
- Wypisywanie się ze spacerów;
- Dodawanie opinii o wyprowadzonym zwierzętach;
- Przeglądanie historii spacerów;
- Przeglądanie profili użytkowników;
- Przeglądanie profili spacerów;
- Zgłaszanie błędów w aplikacji przez użytkowników;
- Zarządzanie kontami użytkowników – blokowanie, banowanie;
- Wyświetlanie bazy danych użytkowników;
- Wyświetlanie bazy danych zwierząt;
- Wyświetlanie bazy danych spacerów;
- System zarządzania zgłoszonymi błędami;

### 2.3.2. Wymagania niefunkcjonalne

- Dostęp do internetu;

## 2.4. Diagram oraz przypadki użycia

### 2.4.1. Przypadki użycia

**Przypadek użycia:** Rejestracja.

**Aktor:** Niezalogowany użytkownik.

**Opis:** Rejestracja w serwisie.

**Warunki wstępne:** Użytkownik niezalogowany, nieposiadający konta, wchodzący do serwisu po raz pierwszy.

**Przebieg:**

- Użytkownik klika w odnośnik ...;
- Użytkownik zostaje przeniesiony do strony z formularzem rejestracyjnym;
- Użytkownik wypełnia dane;
  - Dane są niepoprawne – serwis informuje użytkownika o błędach, przycisk jest nieaktywny;
  - Dane są poprawne – przycisk jest aktywny i użytkownik może stworzyć konto;
- Wysłanie formularza – użytkownik jest informowany o sukcesie bądź błędzie operacji;

**Przypadek użycia:** Logowanie.

**Aktor:** Niezalogowany użytkownik.

**Opis:** Logowanie do serwisu.

**Warunki wstępne:** Użytkownik niezalogowany, posiadający konto w serwisie.

**Przebieg:**

- Użytkownik wypełnia formularz logowania;
- Użytkownik klika w przycisk *Zaloguj się*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;



**Przypadek użycia:** Zmiana zdjęcia profilowego.

**Aktor:** Zalogowany użytkownik.

**Opis:** Zmiana zdjęcia profilowego.

**Warunki wstępne:** Użytkownik zalogowany z aktywnym kontem.

**Przebieg:**

- Użytkownik wchodzi w zakładkę profil;
- Po najechaniu na zdjęcie profilowe klika w ikonę edycji;
- Użytkownik wgrywa nowe zdjęcie;
- Użytkownik klika w przycisk *Zmień zdjęcie*;
- Użytkownik otrzymuje informację zwrotną o statusie wykonanej operacji;

**Przypadek użycia:** Dodanie profilu psa.

**Aktor:** Zalogowany użytkownik.

**Opis:** Dodanie profilu zwierzaka do serwisu.

**Warunki wstępne:** Użytkownik zalogowany, z aktywnym kontem oraz rolą *OWNER*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Dodaj spacer*;
- Użytkownik wypełnia formularz;
  - Dane są poprawne – aktywuje się submit button (przypis);
  - Dane są niepoprawne – pojawia się informacja zwrotna;
- Użytkownik klika w przycisk *Dodaj zwierzaka*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;

**Przypadek użycia:** Dodanie nowego spaceru.

**Aktor:** Zalogowany użytkownik.

**Opis:** Dodanie terminu spaceru.

**Warunki wstępne:** Użytkownik zalogowany, z aktywnym kontem oraz rolą *OWNER*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Dodaj spacer*;
- Użytkownik wypełnia formularz;
  - Dane są poprawne – aktywuje się submit button (przypis);
  - Dane są niepoprawne – pojawia się informacja zwrotna;
- Użytkownik klika w przycisk *Dodaj zwierzaka*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;

**Przypadek użycia:** Przeglądanie historii spacerów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie historii spacerów.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz rolą *OWNER* lub *SITTER*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Historia*;
- Wyświetlana jest lista spacerów w przeszłości;

**Przypadek użycia:** Skomentowanie spaceru.

**Aktor:** Zalogowany użytkownik.

**Opis:** Dodanie komentarza do spaceru.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem, rolą *SITTER* oraz spacerem

w przeszłości.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Historia*;
- Użytkownik wybiera spacer, który chce skomentować;
- Użytkownik wypełnia formularz;
  - Dane są poprawne – aktywuje się submit button (przypis);
  - Dane są niepoprawne – pojawia się informacja zwrotna;
- Użytkownik klika w przycisk *Dodaj komentarz*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;

**Przypadek użycia:** Ocena opiekuna.

**Aktor:** Zalogowany użytkownik.

**Opis:** Dodanie oceny opiekunowi.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem, rolą *OWNER* oraz zakończonym spacerem w przeszłości.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Historia*;
- Użytkownik wybiera spacer z opiekunem, którego chce ocenić;
- Użytkownik wypełnia formularz;
  - Dane są poprawne – aktywuje się submit button (przypis);
  - Dane są niepoprawne – pojawia się informacja zwrotna;
- Użytkownik klika w przycisk *Dodaj ocenę*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;

**Przypadek użycia:** Przeglądanie listy dostępnych spacerów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie listy dostępnych spacerów.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz rolą *SITTER*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Spacery* z aktywnym widokiem wszystkich spacerów;
- Pobierana jest lista psów, bazująca na lokalizacji;
- Lista spacerów wyświetlana jest użytkownikowi;

**Przypadek użycia:** Zapisanie się na spacer.

**Aktor:** Zalogowany użytkownik.

**Opis:** Zapisanie się na spacer.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem, rolą *SITTER* oraz spacer dostępny w przyszłości z wolnym miejscem do zapisu.

**Przebieg: Scenariusz A: Użytkownik znajduje się w zakładce *Spacery***

- Użytkownik klika w przycisk *Zapisz się*;
- Użytkownik jest informowany stosownym komunikatem;

**Scenariusz B: Użytkownik znajduje się w profilu spaceru**

- Użytkownik klika w przycisk *Zapisz się*;
- Użytkownik jest informowany stosownym komunikatem;

**Przypadek użycia:** Wypisanie się ze spaceru.

**Aktor:** Zalogowany użytkownik.

**Opis:** Wypisanie się ze spaceru.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem, rolą *SITTER* oraz spacer w przyszłości na który użytkownik jest już zapisany.

**Przebieg: Scenariusz A: Użytkownik znajduje się w zakładce *Spacery***

- Użytkownik klika w przycisk *Wypisz się*;
- Użytkownik jest informowany stosownym komunikatem;

**Scenariusz B: Użytkownik znajduje się w profilu spaceru**

- Użytkownik klika w przycisk *Wypisz się*;
- Użytkownik jest informowany stosownym komunikatem;

**Przypadek użycia:** Przeglądanie listy nadchodzących spacerów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie listy nadchodzących spacerów na które użytkownik jest zapisany.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem, rolą *SITTER* oraz co najmniej jednym spacerem na który jest zapisany.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Spacery*;
- Użytkownik przełącza widok na *Nadchodzące*;
- Lista nadchodzących spacerów jest wyświetlana użytkownikowi;

**Przypadek użycia:** Przeglądanie terminarza ze spacerami.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie terminarza ze spacerami, na które użytkownik jest zapisany.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz rolą *SITTER*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Terminarz*;
- Użytkownik może wybrać tryb wyświetlania;
  - Miesiąc – widok miesiąca ze zdjęciami psów oraz odnośnikami do profilu spaceru;
  - Tydzień – widok tygodniowy z podstawowymi informacjami o danym spacerze oraz odnośnikiem do profilu spaceru;

**Przypadek użycia:** Przeglądanie profilu użytkownika.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie profilu użytkownika.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz dowolną rolą.

**Przebieg:**

- Użytkownik klika w odnośnik przekierowujący na profil innego użytkownika;
- Wyświetlają się szczegółowe informacje o użytkowniku;

**Przypadek użycia:** Przeglądanie profilu psa.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie profilu psa.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz dowolną rolą.

**Przebieg:**

- Użytkownik klika w odnośnik przekierowujący na profil psa;
- Wyświetlają się szczegółowe informacje;

**Przypadek użycia:** Przeglądanie profilu spaceru.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie profilu spaceru.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz dowolną rolą.

**Przebieg:**

- Użytkownik klika w odnośnik przekierowujący na profil spaceru;
- Wyświetlają się szczegółowe informacje o spacerze;

**Przypadek użycia:** Zgłaszanie błędów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Zalogowany użytkownik.

**Warunki wstępne:** Zalogowany użytkownik z aktywnym kontem oraz dowolną rolą.

**Przebieg:**

- Użytkownik klika w ikonę błędu w prawym dolnym rogu;
- Użytkownik wypełnia formularz;
  - Dane są poprawne – aktywuje się submit button (przypis);
  - Dane są niepoprawne – pojawia się informacja zwrotna;
- Użytkownik klika w przycisk *Zgłoś błąd*;
  - W przypadku sukcesu użytkownik zostaje przekierowany do strony aplikacji;
  - W przypadku błędu użytkownik jest informowany o błędzie;

**Przypadek użycia:** Przeglądanie listy wszystkich użytkowników.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie bazy użytkowników.

**Warunki wstępne:** Zalogowany użytkownik z rolą *ADMIN*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Użytkownicy*;
- Pojawia się lista wszystkich użytkowników zarejestrowanych w serwisie;

**Przypadek użycia:** Przeglądanie listy wszystkich spacerów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie bazy spacerów.

**Warunki wstępne:** Zalogowany użytkownik z rolą *ADMIN*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Zwierzęta i spacery*;
- Użytkownik wybiera widok spacerów;
- Pojawia się lista wszystkich spacerów dodanych w serwisie;

**Przypadek użycia:** Przeglądanie listy wszystkich psów .

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie bazy zwierząt.

**Warunki wstępne:** Zalogowany użytkownik z rolą *ADMIN*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Zwierzęta i spacery*;
- Użytkownik wybiera widok zwierząt;
- Pojawia się lista wszystkich psów dodanych w serwisie;

**Przypadek użycia:** Przeglądanie aktywności użytkowników.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie aktywności użytkowników.

**Warunki wstępne:** Zalogowany użytkownik z rolą *ADMIN* .

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Aktywność*;
- Pojawia się lista aktywności użytkowników na serwisie;

**Przypadek użycia:** Przeglądanie listy zgłoszonych błędów.

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie listy zgłoszonych błędów przez użytkowników.

**Warunki wstępne:** Zalogowany użytkownik z rolą *ADMIN*.

**Przebieg:**

- Użytkownik wchodzi w zakładkę *Błędy*;
- Pojawiają się lista błędów podzielona na 3 grupy – nowe, w trakcie realizacji, naprawione;

**Przypadek użycia:** Przeglądanie dashboardu (przypis).

**Aktor:** Zalogowany użytkownik.

**Opis:** Przeglądanie dashboardu.

**Warunki wstępne:** Zalogowany użytkownik z rolą *OWNER* lub *SITTER*.

**Przebieg:**

- Użytkownik znajduje się w zakładce dashboard;
- Pojawia się lista nadchodzących spacerów oraz powiadomień;

## 2.5. Widoki, mockupy

Kluczowym elementem przy projektowaniu serwisów internetowych jest stworzenie prototypów widoków aplikacji. Ten zabieg znacząco ułatwia i przyspiesza pracę w etapie implementowania warstwy dostępowej.

Do stworzenia prototypów użyto oprogramowania *Adobe XD*. Jest to popularne narzędzie do tworzenia interfejsów użytkownika. Program oferuje szereg narzędzi do projektowania wyglądu aplikacji. Zaletą używania programów, których przeznaczeniem jest kreowanie makiet aplikacji internetowych, jest zastosowanie właściwości dodawanych elementów znanych z arkuszy stylów takich, jak *CSS* bądź *SCSS*. Pozwala to w późniejszym czasie bezproblemowo przenosić zaprojektowane wyglądy do aplikacji frontendowej.

## 2.6. Backend

Każda aplikacja internetowa posiada swój silnik, który jest określany jako *backend*. Jest to logika biznesowa, stojąca po stronie serwera. Odpowiada za pobieranie oraz przetwarzanie danych, komunikację z bazą danych oraz w większym stopniu za bezpieczeństwo całej aplikacji. Aplikacje backendowe udostępniają zbiór funkcjonalności, inaczej zwane REST API (*ang. ...*), które są wykorzystywane do transferu danych między zapleczem technicznym a warstwą dostępową aplikacji internetowej.

W omawianej aplikacji do implementacji backendu wykorzystano język *Java* - popularny język obiektowy oparty na klasach, charakteryzujący się kodem, który po kompilacji można uruchomić na każdej platformie. Niezbędny był również framework *Spring Boot*, dzięki któremu możliwe było stworzenie rozbudowanego API aplikacji. *Spring Boot* jest narzędziem opartym o framework *Spring*, zapewniającym dodatkową konfigurację oraz kontener aplikacji. Takie zastosowanie pozwala na szybsze tworzenie aplikacji bez konieczności dodatkowej konfiguracji. Użyty framework korzysta z mechanizmu wstrzykiwania zależności (*ang. Dependency Injection*). *Dependency injection* odpowiada za tworzenie i dostarczanie obiektów w odpowiednim momencie do danej klasy. Obecnie *Java* w połączeniu ze *Spring Boot*em jest powszechnie używanym narzędziem wśród wielu firm na całym świecie.

Do tworzenia warstw biznesowych przy wykorzystaniu wyżej wymienionych narzędzi, niezbędnym elementem jest zbiór zależności. Pozwalają rozszerzyć bazowe funkcjonalności o nowe, skonkretyzowane metody i rozwiązania. W przypadku aplikacji, która jest tematem

niniejszej pracy wykorzystano następujące zależności:

**Spring Web** – wprowadza do projektu podstawowe funkcje integracyjne, inicjalizuje kontener IoC, zorientowuje kontekst aplikacji na sieć oraz dodaje klienta HTTP.

**Spring Security** – konfigurowalna platforma zapewniająca kontrolę nad dostępem do aplikacji. Stanowi standard zabezpieczeń oraz pozwala na uwierzytelnianie użytkowników.

**JSON Web Token** – popularne rozwiązanie do uwierzytelniania użytkowników. Wykorzystywany jako token autoryzacyjny, generowany po stronie backendu i przechowywany po stronie frontendu, umożliwia na dostęp do zasobów API aplikacji.

**MongoDB Driver** – zależność umożliwiająca skonfigurowanie aplikacji z bazą danych. Udostępnia szereg metod umożliwiających wykonywanie operacji z przetwarzaniem danych w bazie.

**SwaggerUI** – dostarcza wizualną reprezentację oraz dostęp do punktów końcowych (*ang. endpoint*) aplikacji. Umożliwia na testowanie zaimplementowanych funkcjonalności bezpośrednio po stronie backendu.

**Lombok** – jest procesorem adnotacji, umożliwiającym na generowanie kodu przed kompilacją po zastosowaniu odpowiedniej adnotacji nad klasą. Wykorzystywany jest przy obiektach do generowania powtarzalnych metod takich jak konstruktory, gettery oraz settery.

W trakcie implementacji technicznej strony aplikacji niezbędnym etapem jest wybór bazy danych dla serwisu. Obecnie od wielu lat wykorzystywane są relacyjne bazy danych. Jednak w ciągu ostatnich lat można zauważyć rosnącą popularność baz nierelacyjnych. Wśród nich w czołówce plasuje się MongoDB. Charakterystycznymi cechami omawianej bazy danych jest zrezygnowanie z relacji między kolekcjami, operacja na dokumentach oraz przechowywanie danych w formacie BSON - jest to format JSON w postaci binarnej. Mongo, tak jak wiele innych baz NoSQL nie posiada żadnych reguł implementacyjnych co pozwala na dowolność w kwestii przechowywania danych. Jest to również nowoczesne podejście do modelowania baz danych. Dlatego też zdecydowano się na zastosowanie MongoDB jako bazy danych dla omawianej aplikacji.

Warstwa biznesowa aplikacji została zaimplementowana przy użyciu języka Java oraz frameworka Spring. Połączenie tych dwóch narzędzi pozwala stworzyć rozbudowany silnik aplikacji zwany API (*ang. Application Programming Interface*) Java jest wysokopoziomowym, obiektowym językiem, często stosowanym przez firmy IT do budowania warstw backendowych lub aplikacji mobilnych.

## 2.7. Frontend

Do komunikacji między użytkownikami a backendem używa się warstw dostępowych, znanych również w środowisku programistycznym jako frontend. Frontend służy do odbierania danych od użytkowników, przetwarzania ich do odpowiedniego formatu, wymaganego przez API oraz wysyłanie danych do backendu. W obecnych czasach dużą popularnością cieszą się frameworki oparte na języku JavaScript - języku skryptowym który umożliwia wprowadzenie funkcjonalności do plików HTML. Takie frameworki pozwalają na tworzenie aplikacji w technice SPA.

Frontend został zaimplementowany przy wykorzystaniu frameworka *Angular*, który obecnie znajduje się w czołówce popularności. Pozwala na tworzenie wydajnych aplikacji SPA. Charakterystyczną cechą omawianego frameworka jest wykorzystanie komponentów – mogą być składowymi lub definiować całe widoki z dołączoną logiką do obsługi danych. Angular wykorzystuje również mechanizm *dependency injection*, który jest niezbędny do tworzenia modułowych komponentów, które można wielokrotnie używać. Znaczącą rolę w działaniu aplikacji pełnią moduły. Są to zarówno zewnętrzne biblioteki, posiadające szereg rozwiązań

implementacyjnych oraz komponenty tworzone przez programistę. W trakcie implementacji warstwy dostępowej zastosowano zbiór zewnętrznych modułów oraz bibliotek, zapewniających prawidłowe działanie aplikacji.

**BrowserModule** – odpowiada za uruchamianie aplikacji w przeglądarce.

**AppRoutingModule** – dodaje możliwość nawigacji z uwzględnieniem odpowiednich adresów URL.

**ReactiveFormsModule** – odpowiada za tworzenie reaktywnych formularzy.

**HttpClientModule** – pozwala na komunikację poprzez protokół HTTP z API.

**RxJS** – biblioteka wprowadzająca możliwość programowania reaktywnego w aplikacji.

Doświadczenia wizualne płynące z korzystania z serwisów internetowych są w obecnych czasach niezwykle istotne – można by założyć że atrakcyjność wyglądu góruje nad funkcjonalnościami aplikacji. Dlatego też bardzo ważnym elementem przy tworzeniu frontendowych aplikacji jest zadbanie o odpowiedni wygląd aplikacji poprzed dołożenie do projektu arkuszy stylów. Najpopularniejszym i najczęściej używanym obecnie jest *CSS*. Pozwala w prosty sposób tworzyć selektory klas i stylizować poszczególne elementy składające się na cały widok. W trakcie implementacji makiet widoków zdecydowano się na użycie bardziej rozbudowanego narzędzia do tworzenia arkuszy stylów – *SCSS*. Posiada ten sam zbiór właściwości które można stylizować, natomiast rozszerza bazową wersję o możliwość definiowania zmiennych oraz tworzenie zagnieżdżonych selektorów. Takie dodatkowe funkcjonalności pozwalają tworzyć mniej rozbudowane, prostsze arkusze stylów, które w znaczący sposób są podatne na późniejsze modyfikacje.

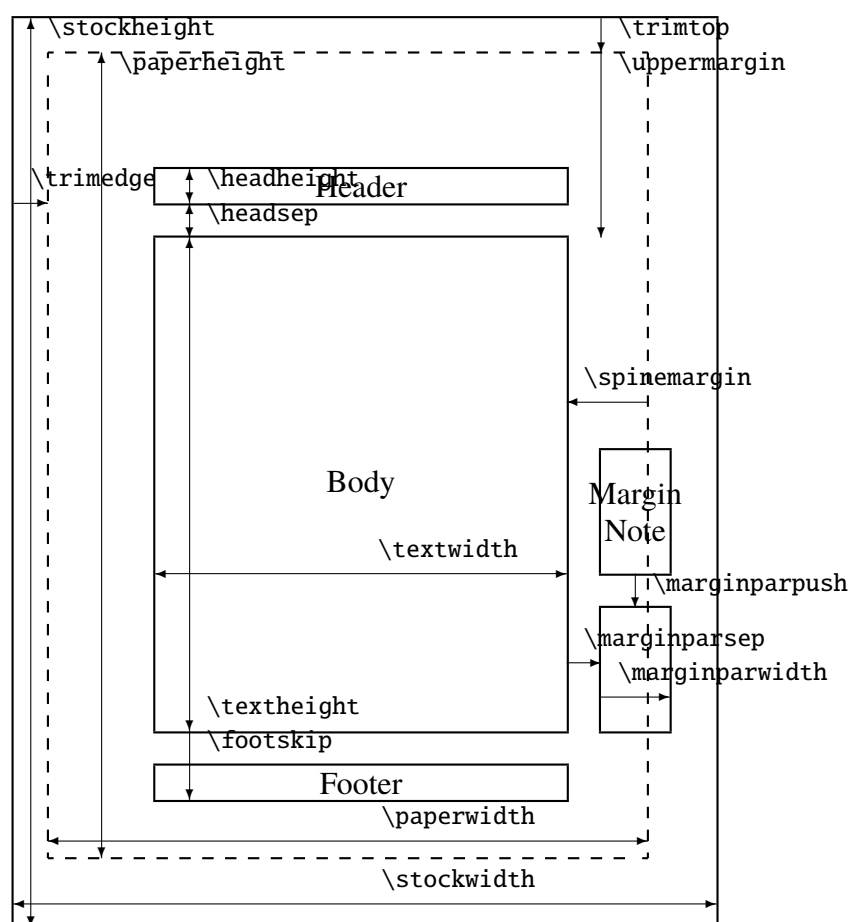
## Rozdział 3

# Zalecenia dotyczące formatowania

### 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.

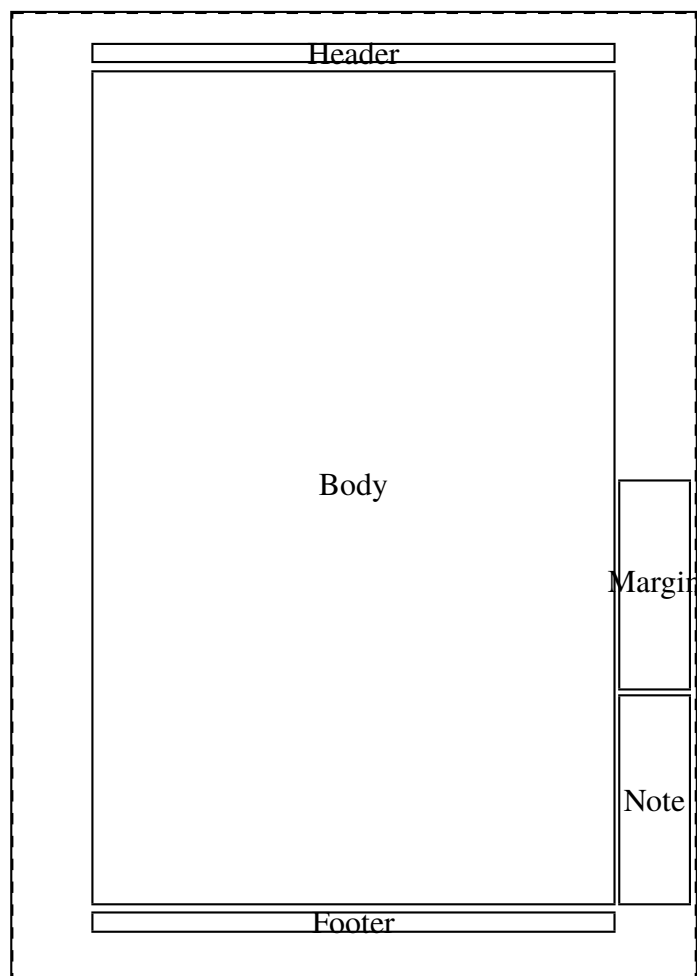
Dashed lines represent the actual page size after trimming the stock.



Rys. 3.1: Układ strony nieparzystej dla dokumentu klasy memoir



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

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{\spinemargin}{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 (zgrubnie) położenie słów kluczowych i streszczenia. 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 wzorzec strony tytułowej (używany od lat) oraz podano wymagania odnośnie wzorca z logotypu uczelnianego.

Rys. 3.3: Oficjalny szablon strony tytułowej pracy dyplomowej, <http://www.logotyp.pwr.edu.pl/Default.aspx?page=PracaDyplomowa> [dostęp dnia 20.04.2016]

Wymagania co do wielkości znaków na stronie tytułowej są następujące:

- według uczelnianego logotypu

Nazwa jednostki organizacyjnej: Garamond 16 pt  
 Napis "PRACA DYPLOMOWA 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 DYPLOMOWA (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)  
 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 „l” 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 `txfonts`, który należy wcześniej zainstalować (`MikTeX` zainstaluje go automatycznie podczas pierwszej kompilacji szablonu).

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

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	<b>Rozdział 1</b>	<code>\huge \bfseries</code>	25pt	30pt
Tytuł rozdziału	<b>Wstęp</b>	<code>\Huge \bfseries</code>	30pt	37pt
Nr i tytuł sekcji	<b>1.1. Wprowadzenie</b>	<code>\Large \bfseries</code>	17pt	22pt
Nr i tytuł podsekcji	<b>1.1.1. Cel szczegółowy</b>	<code>\large \bfseries</code>	14.5pt	18pt
Tytuł podpodsekcji	<b>Założenia</b>	<code>\normalsize \bfseries</code>	12pt	14.5pt
Tytuł paragrafu	<b>Podstawy</b> Opis ...	<code>\normalsize \bfseries</code>	12pt	14.5pt
Tekst wiodący	Niniejszy dokument ...	<code>\normalsize</code>	12pt	14.5pt
Nagłówek strony	3.2. Czcionka wiodąca ...	<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

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

```

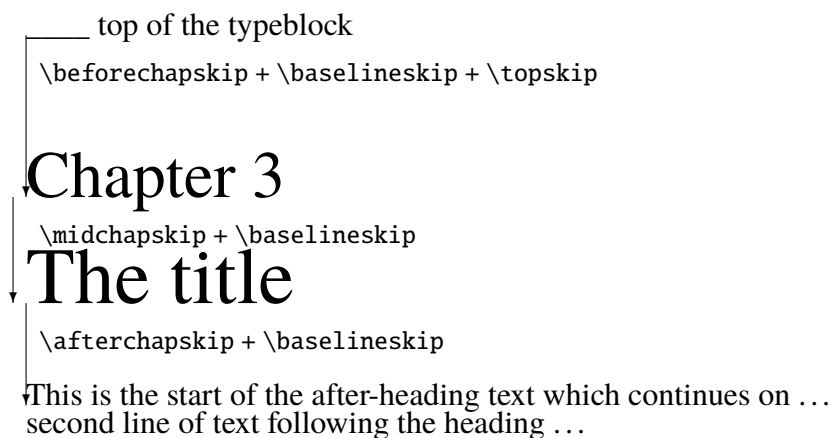
Aby wykorzystać te czcionki poza systemem LaTeX, wystarczy pobrać je spod adresów (ważnych na dzień 1.04.2016): <https://www.ctan.org/tex-archive/fonts/cm/ps-type1/bakoma/ttf/?lang=en>, <http://www.gust.org.pl/projects/e-foundry/latin-modern>, <http://www.gust.org.pl/projects/e-foundry/tex-gyre>, <https://bitbucket.org/georgd/eb-garamond/downloads>, 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

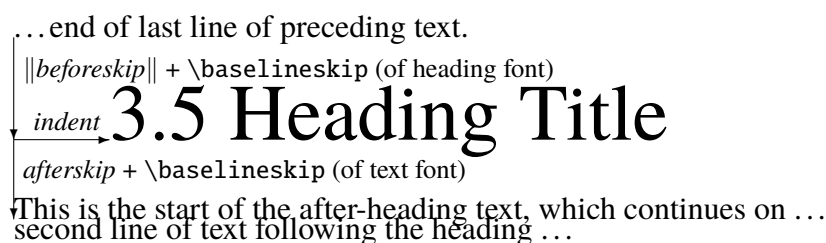
Każdy rozdział pracy powinien rozpoczynać się od nowej strony. Jej wygląd powinien być kontrolowany parametrami pokazanymi na rysunku 3.4. W niniejszym szablonie (dokument klasy memoir z opcją [12pt]) przyjęto następujące wartości tych parametrów:

- `\beforechapskip` (50.0pt) + `\baselineskip` of `\huge` (30pt) + `\topskip` (12.0pt) = 92pt (3.246cm)
- `\midchapskip` (20.0pt) + `\baselineskip` of `\Huge` (37pt) = 57 pt (2.011cm)
- `\afterchapskip` (40.0pt) + `\baselineskip` of `\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,



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



Rys. 3.5: Kontrola ustawień odległości w tytułach kolejnych sekcji

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 (gdyby 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 `beforesecskip` 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
- `beforesecskip` = -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
- `inttextsep` = 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.

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 należy 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 tilde, 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 italic wraz ze skrótem oznaczającym język, w szczególności ma to zastosowanie przy rozwijaniu skrótów, np. OGC (ang. *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).

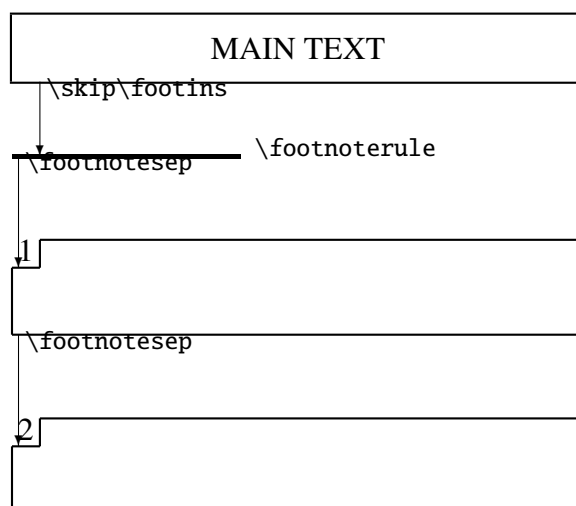
## 3.5. Opisy tabel i rysunków

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 <sup>1</sup>). 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
```



Rys. 3.6: Parametry sterujące przypisami dolnymi

### 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
\@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*}.
```

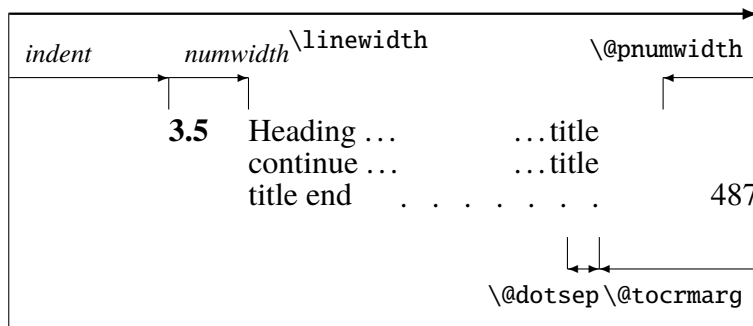
<sup>1</sup>Tekst przypisu

```

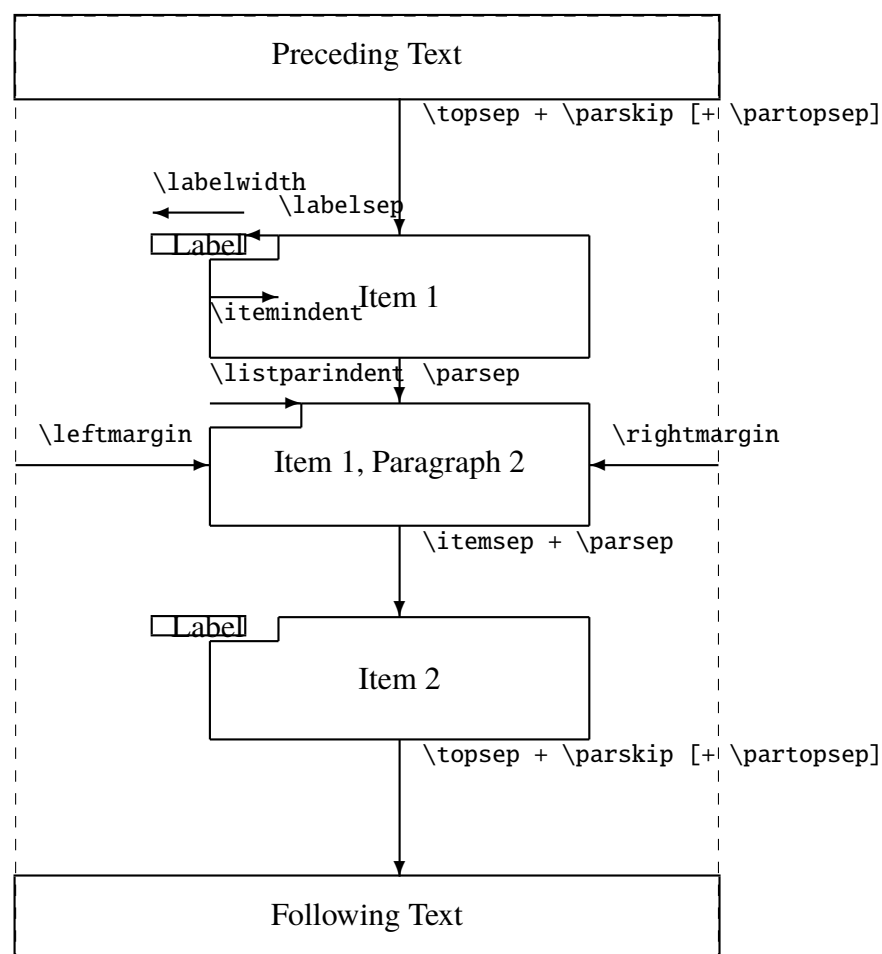
\setlistdepth{4}
\setlist[itemize,1]{label=$\bullet$}
\setlist[itemize,2]{label=\normalfont\bfseries\textendash}
\setlist[itemize,3]{label=$\ast$}
\setlist[itemize,4]{label=$\cdot$}
\renewlist{itemize}{itemize}{4}

```

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



Rys. 3.7: Parametryzacja wyglądu spisu treści



Rys. 3.8: Parametryzacja list wyliczeniowych i wypunktowań



### 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).

$$\int_0^{10\nu\sum i} x dx \tag{3.1}$$

# Rozdział 4

## Redakcja pracy

### 4.1. Układ pracy

Standardowo praca powinna być zredagowana w następującym układzie:

Strona tytułowa
Strona z dedykacją (opcjonalna)
Spis treści
Spis rysunków (opcjonalny)
Spis tabel (opcjonalny)
Skróty (wykaz opcjonalny)
1. Wstęp
1.1 Cel i zakres pracy
1.2 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
...
\$. Zawartość płyty CD/DVD
Indeks rzeczowy (opcjonalny)

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).

Wstęp – pierwszy rozdział, w którym powinien znaleźć się opis dziedziny, w jakiej osadzona jest praca, oraz wyjaśnienie motywacji do podjęcia tematu. 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. 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.

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. W przypadku korzystania z szablonu odstępny 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}
```

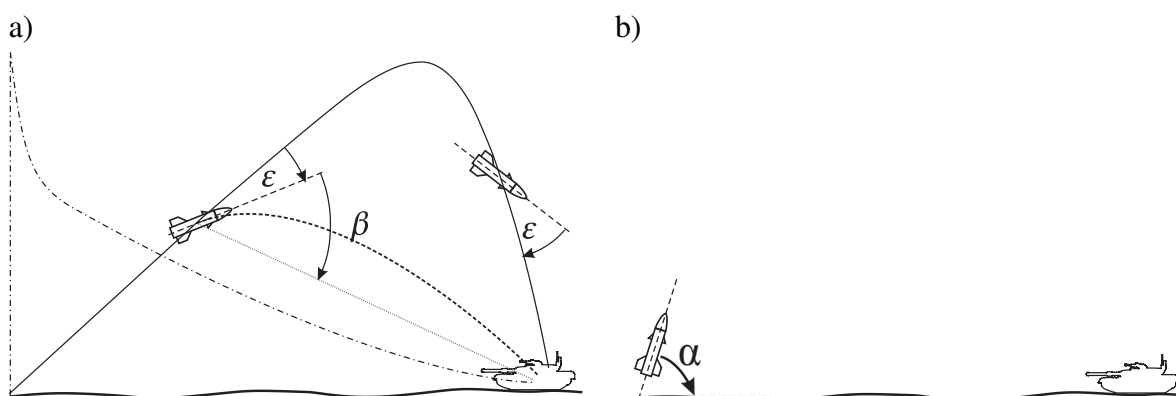
```

\begin{figure}[htb]
\centering
\begin{tabular}{@{}ll@{}}
a) & b) \\
\includegraphics[width=0.475\textwidth]{rys05/alfa1} & \\
\includegraphics[width=0.475\textwidth]{rys05/beta1} & \\
\end{tabular}
\caption{Wyznaczanie trajektorii lotu rakiety:  
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. 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.

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{\vskip3ex\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).

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 przestrzenie 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 odwzorowaniem 1:1 domyślnego układu obserwowanych elementów. Ważne jest, by na zrzucie z ekranu pokazać interesujący, opisywany fragment i żeby ten fragment był czytelny.

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.

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!!!

## 5.2. 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 linijce 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=`).

## 5.3. 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 dokumentacji `.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 [3, 1]” (gdzie zastosowano komendę `\cite{JS07,SQL2}`).

Co do zawartości rekordów bibliograficznych - style `bibtex`owe 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}
}
```

## 5.4. 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 \textit{Świat} \LaTeX, 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`).

## 5.5. 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 „\textit{jak tutaj}”, tylko że czasem trudno pisać się 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
\doublespacing

```

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 -dBATCH
-sOutputFile=Dyplom-compressed.pdf Dyplom.pdf

```

# Rozdział 6

## Podsumowanie

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.

### 6.1. Sekcja poziom 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 poziom 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 poziom 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 poziom 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. [2]

# Spis rysunków

3.1. Układ strony nieparzystej dla dokumentu klasy memoir . . . . .	16
3.2. Rzeczywisty układ strony nieparzystej w tym dokumencie . . . . .	17
3.3. Oficjalny szablon strony tytułowej pracy dyplomowej, <a href="http://www.logotyp.pwr.edu.pl/Default.aspx?page=PracaDyplomowa">http://www.logotyp.pwr.edu.pl/Default.aspx?page=PracaDyplomowa</a> [dostęp dnia 20.04.2016] .	18
3.4. Parametry sterujące wielkościami odstępów na stronie z tytułem rozdziału . . . . .	21
3.5. Kontrola ustawień odległości w tytułach kolejnych sekcji . . . . .	21
3.6. Parametry sterujące przypisami dolnymi . . . . .	23
3.7. Parametryzacja wyglądu spisu treści . . . . .	24
3.8. Parametryzacja list wyliczeniowych i wypunktowań . . . . .	24
5.1. Dwa znaki kanji – giri . . . . .	29
5.2. Wyznaczanie trajektorii lotu rakiety: a) trzy podejścia, b) podejście praktyczne . .	29

# Spis tabel

- 3.1. Zestawienie czcionek elementów podziału dokumentu, tekstu wiodącego, nagłówka i stopki oraz podpisów (Rozm. – rozmiar czcionki, Odst. – baselineskip) . . . . 20

# 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] O. M. Group. Business Process Model and Notation (BPMN) Version 2.0, Sty. 2011. <http://www.omg.org/spec/BPMN/2.0> [dostęp dnia 20 czerwca 2015].
- [3] J. Jędrzejczyk, B. Śródka. Segmentacja obrazów metodą drzew decyzyjnych. Raport instytutowy, Politechnika Wrocławska, Wydział Elektroniki, 2007.

# **Dodatek A**

## **Tytuł dodatku**

Zasady przyznawania stopnia naukowego doktora i doktora habilitowanego w Polsce określa ustawa z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz.U. nr 65 z 2003 r., poz. 595 (Dz. U. z 2003 r. Nr 65, poz. 595)). Poprzednie polskie uregulowania nie wymagały bezwzględnie posiadania przez kandydata tytułu zawodowego magistra lub równorzędnego (choć zasada ta zazwyczaj była przestrzegana) i zdarzały się nadzwyczajne przypadki nadawania stopnia naukowego doktora osobom bez studiów wyższych, np. słynnemu matematykowi lwowskiemu – późniejszemu profesorowi Stefanowi Banachowi.

W innych krajach również zazwyczaj do przyznania stopnia naukowego doktora potrzebny jest dyplom ukończenia uczelni wyższej, ale nie wszędzie.

## **Dodatek B**

# **Opis załączonej płyty CD/DVD**

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



# Indeks rzeczowy

\*, 32

$\mathcal{C}$ , 32

generowanie

– indeksu, 32

– wykazu literatury, 32

linia komend, 32

Światło, 32