

Spis treści

1. Układ i zawartość pracy	2
1.1. Wstęp	2
1.2. Opis dziedziny problemu	3
1.3. Wykorzystane technologie	3
1.4. Dokumentacja techniczna	4
1.5. Dokumentacja użytkownika	4
1.6. Bibliografia i spisy	4
2. Uwagi redakcyjne	5
2.1. Formatowanie tekstu	5
2.2. Tabele, rysunki, listingi, odwołania	5
2.3. Forma wypowiedzi w pracy	7
2.4. Edycja tekstu w \LaTeX	7
Bibliografia	8
Spis rysunków	9
Spis tabel	10

1. Układ i zawartość pracy

Praca inżynierska powinna się składać z czterech rozdziałów. Dwa pierwsze rozdziały zawierają opis dziedziny problemu i opis wykorzystywanych technologii (jedno zagadnienie w każdym z rozdziałów). Dwa kolejne rozdziały stanowią dokumentację techniczną powstałej aplikacji oraz dokumentację użytkownika.

1.1. Wstęp

Wstęp pracy dyplomowej powinien być podzielony na dwie części. Pierwsza część zawiera opis problemu:

- jaki jest obecny 'stan faktyczny'?
- jaki problem dostrzegamy?
- jakie są istniejące rozwiązania?
- dlaczego warto się tym zajmować?

Następnie wskazuje się **cel** i **zakres** pracy. Celem będzie najczęściej rozwiązanie problemu poprzez dostarczenie aplikacji, która wspomaga jakieś działanie. Zakres informuje, co właściwie zostało w ramach pracy zrobione.

W drugiej części wstępu opisuje się zawartość rozdziałów, na przykład w taki sposób:

W rozdziale pierwszym niniejszej pracy opisano najważniejsze pojęcia z dziedziny... Rozdział drugi zawiera opis technologii wykorzystanych w niniejszej pracy. Dokumentację techniczną powstałej aplikacji MyAwesomeApp przedstawiono w rozdziale 3. W rozdziale 4 zawarto dokumentację użytkownika aplikacji MyAwesomeApp.

UWAGI:

- wstęp do pracy piszemy na końcu - dla uniknięcia ciągłych jego przeróbek
- wstęp powinien mieć ok **1.5 strony**

1.2. Opis dziedziny problemu

W pierwszym rozdziale pracy powinno znaleźć się wprowadzenie w dziedzinę problemu, którego dotyczy praca. Należy opisać na przykład:

- Postawienie problemu: wprowadzenie do tematu, jak dużo ludzi jest związanych z dziedziną pracy, jak obecnie wyglądają rozwiązania z dziedziny problemu (jakie cechy mają), jakich cech nie mają. W ten sposób dochodzimy do *postawienia problemu*. Potem wskazujemy *pomysł* na rozwiązanie.
- Szczegółowy opis dziedziny problemu: najważniejsze pojęcia, zasady, reguły itd.
- Przegląd istniejących rozwiązań: jakie rozwiązania są obecnie na rynku, szczegółowy opis ich funkcjonalności, ceny i wymagania sprzętowe tych rozwiązań, dostępność, rysunki z działania (o ile da się je zrobić)
- Podsumowanie - uzasadnienie, dlaczego wybrane do implementacji rozwiązanie jest sensowne i jak poradzi sobie z problemem.

UWAGA: Rozdział powinien zawierać kilkudzaniowe wprowadzenie.

1.3. Wykorzystane technologie

W drugim rozdziale pracy (można zamienić kolejność, jeżeli praca dotyczy technologii a tworzona aplikacja jest tylko przykładem jej wykorzystania). Należy napisać:

- Bardzo krótkie streszczenie dotyczące aktualnie dostępnych technologii (np. w zakresie aplikacji internetowych należy wymienić kilka serwerów aplikacji, frameworków do logiki biznesowej i do interfejsu użytkownika). Należy przyjąć pewne założenia pozwalające ograniczyć wybór spośród dużej liczby technologii. Najlepiej oczywiście aby były one sensowne.
- Szczegółowe przedstawienie technologii konkurencyjnych, np. jeśli piszemy pracę z dziedziny frameworków Javy do tworzenia aplikacji internetowych, to konkurencją najczęściej będą inne frameworki Javy do tworzenia aplikacji internetowych. Zakres konkurencji wynika z przyjętych wcześniej założeń. Nie należy jednak tworzyć sztucznych ograniczeń aby tylko 'nasza' technologia się nadawała.
- Szczegółowy opis wybranej technologii. Podrozdział powinien zawierać: opis architektury frameworka, najważniejsze cechy, opis konstrukcji składniowych (z przykładowym kodem na listingach).

1.4. Dokumentacja techniczna

Rozdział powinien stanowić opis, w jaki sposób stworzona aplikacja została wykonana. Na początku powinien znajdować się *opis założeń*, aby była możliwa późniejsza weryfikacja czy zaproponowana architektura aplikacji 'jest sensowna'. Następne podrozdziały mogą być związane z warstwami (np. dane, logika, widok lub model, kontrolery, widoki). W tym dokumentacji technicznej nie powinno się już dokładać nowych wiadomości z zakresu samej technologii, w której wykonana jest aplikacja - jeżeli taki opis wyda się niezbędny na tym etapie pisania pracy, to należy dodać odpowiednią treść do rozdziału o technologii.

W rozdziale mile widziane są diagramy. Większość diagramów można wygenerować automatycznie przy pomocy środowiska deweloperskiego. W szczególności:

- diagram komponentów i wdrożeń, diagramy klas dla poszczególnych warstw
- diagram sekwencji, stanów, czynności, w zależności od potrzeby

W tym rozdziale mile widziane są również listingi pokazujące wykorzystanie wybranych aspektów stosowanej technologii. Nie powinny być one zbyt długie, ani stanowić powielenia 'przykładowych' listingów z rozdziału o technologii.

1.5. Dokumentacja użytkownika

W ostatnim rozdziale pracy należy opisać funkcjonalność dostarczonej aplikacji. Najlepiej pośilkować się w tym miejscu diagramem przypadków użycia. W kolejnych podrozdziałach należy zawrzeć opis typowego scenariusza korzystania ze stworzonej aplikacji. Bardzo mile widziane są rysunki przedstawiające wygląd aplikacji po realizacji kolejnych kroków scenariusza.

1.6. Bibliografia i spisy

Praca powinna zawierać:

- spis treści na początku
- bibliografię (na końcu)
- spis rysunków (na końcu)
- spis tabel (na końcu)
- spis listingów (na końcu)

2. Uwagi redakcyjne

2.1. Formatowanie tekstu

Wygodna czcionka do czytania to Times New Roman (12pt), interlinia 1.5. Do wydruku jednostronnego lewy margines powinien być szerszy o ok. 1cm - ułatwi to oprawę pracy. Tekst powinien być wyjustowany a akapit rozpoczynany od wcięcia.

- Wprowadzane pojęcia piszemy **pogrubioną** czcionką.

PRZYKŁAD: Chromosomy osobników wchodzących w skład populacji składają się z **genów**. Każdy gen przechowuje zakodowaną informację o ...

- Wszelkie skróty powinny być wyjaśnione.

PRZYKŁAD: W implementacji projektu MyAwesomeApp wykorzystano metodykę DDD (Domain Driven Design).

- Pojęcia tłumaczone z języka angielskiego powinny mieć podane w nawiasie oryginalne brzmienie zapisane *kursywą*.

PRZYKŁAD: Wstrzykiwanie zależności (ang. *dependency injection*) jest techniką pozwalającą na odseparowanie komponentów wchodzących w skład aplikacji.

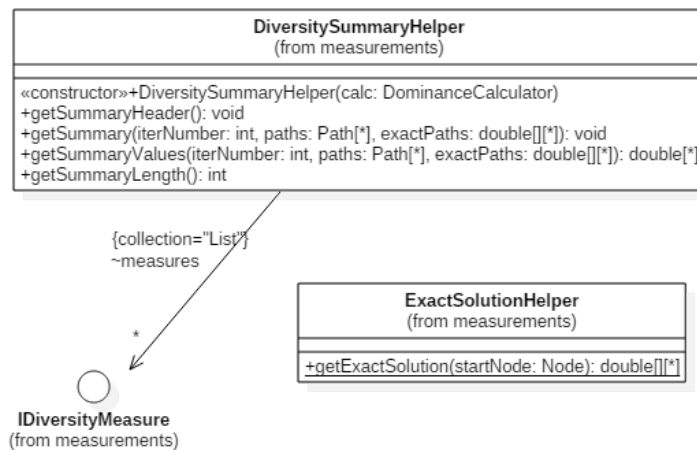
Oczywiście powyższe zasady stosujemy wyłącznie w przypadku pierwszego pojawienia się danego pojęcia w tekście.

2.2. Tabele, rysunki, listingi, odwołania

Rysunki i tabele powinny być wycentrowane. Podpisy do rysunków i tabel wstawiamy według poniższego przykładu:

Tabela 2.1. Przykładowe wartości dla funkcji $x^2 + y^2$

x	y	wartość
0	0	0
2	1	5
3	3	18



Rysunek 2.1. Diagram klas dla komponentu [Opracowanie własne]

Listingi powinny być numerowane. Kod w listingach powinien być pisany czcionką o stałej szerokości z zachowaniem wcięć takich jakie stosuje się w IDE do jego pisania. Można usunąć kolorowanie składni, jeżeli przy kopiowaniu ze środowiska developerskiego zostało ono zachowane. Bardzo wygodnie jest, kiedy linie na listingach są numerowane (wtedy można się do nich odwoływać). Najważniejsze linie można pogrubić.

Podpisy do tabel powinny być wyśrodkowane i położone nad tabelą. Podpisy do rysunków powinny być wyśrodkowane i położone pod rysunkiem. Poniżej pokazano sposób odwoływania się do tabel i rysunków. Analogicznie odwołujemy się do rozdziałów.

Przykładowe wartości funkcji $x^2 + y^2$ pokazano w tabeli 2.1.

... obliczono przykładowe wartości funkcji (Tabela 2.1).

W rozdziale 1 omówiono podstawowe zasady pisania pracy dyplomowej.

Odwołania do literatury powinny być zrealizowane jak w poniższym przykładzie (wpisy bibliograficzne pod koniec):

Pojęcie optymalizacji wielokryterialnej zostało szczegółowo opisane w pracy [4]. Dhillon [1] wprowadził pojęcie metod spektralnych podziału grafu. Zagadnienie to jest opisane również w pracach [3, 2].

2.3. Forma wypowiedzi w pracy

- Praca powinna być napisana w formie bezosobowej, na przykład: *'uzyskane wyniki pokazują, że'* zamiast *'uzyskałam/em wyniki, które pokazują, że'* lub *'opracowana aplikacja'* zamiast *'moja aplikacja'*
- Do rozdziałów/rysunków/tabel odwołujemy się przy pomocy numerów a nie słów *'powyżej'*, *'poniżej'*, *'w poprzednim rozdziale'*.
- Pisząc o samej pracy używamy zwrotu *'niniejsza praca'*, bieżący rozdział to *'niniejszy rozdział'*
- Nie korzystamy ze skrótów, takich jak *np.*, *nr*, *itd.*
- Staramy się pisać możliwie krótkimi zdaniami - zdania wielokrotnie złożone są bardzo trudne do czytania.
- W czasie pisania nie zmieniamy czasu, strony ani trybu czasownika. Szczególnie razi to w następujących po sobie zdaniach.
- Każdy rozdział zaczynamy kilkoma zdaniami wprowadzenia - *'W niniejszym rozdziale zaprezentowano zbiór najważniejszych pojęć z dziedziny W kolejnych podrozdziałach omówiono ...'*
- Każdy rozdział kończymy kilkoma zdaniami podsumowania/wniosków

2.4. Edycja tekstu w L^AT_EX

Gorąco zachęcam do składania pracy w L^AT_EX. Ma to cały zestaw zalet:

- Bardzo trudno jest złożyć pracę brzydko.
- Wszelkie odwołania realizowane są półautomatycznie.
- Praca ze środowiskiem do L^AT_EX-a bardzo przypomina pracę z edytorem tekstu, może za wyjątkiem tego, że nie trzeba co chwilę odrywać rąk od klawiatury i wybierać czegoś z menu.
- Składnia jest bardzo prosta - na użytek pracy inżynierskiej potrzebna jest znajomość co najwyżej kilku komend.
- Jest gotowy szablon do pracy
- Tekst pracy można wersjonować

Aby edycja tekstu była łatwiejsza i przyjemniejsza polecam korzystanie z narzędzia **TeXstudio** i dystrybucji **TeXlive**.

Bibliografia

- [1] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001. (Cytowanie na stronie 6.)
- [2] Diansheng Guo. Regionalization with dynamically constrained agglomerative clustering and partitioning (redcap). *International Journal of Geographical Information Science*, 22(7):801–823, 2008. (Cytowanie na stronie 6.)
- [3] Leszek Rutkowski, Marcin Korytkowski, Rafal Scherer, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek M. Zurada, editors. *Artificial Intelligence and Soft Computing - 11th International Conference, ICAISC 2012, Zakopane, Poland, April 29-May 3, 2012, Proceedings, Part II*, volume 7268 of *Lecture Notes in Computer Science*. Springer, 2012. (Cytowanie na stronie 6.)
- [4] Arthur Warburton. Approximation of pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1):70–79, 1987. (Cytowanie na stronie 6.)

Spis rysunków

2.1. Diagram klas dla komponentu [Opracowanie własne]	6
---	---

Spis tabel

2.1. Przykładowe wartości dla funkcji $x^2 + y^2$	5
---	---