

Uniwersytet Mikołaja Kopernika w Toruniu
Wydział Fizyki, Astronomii i Informatyki Stosowanej

<Imię i nazwisko>

nr albumu: <numer albumu>

Informatyka stosowana

Praca inżynierska

<Tytuł pracy dyplomowej>

Opiekun pracy dyplomowej

<tytuł, stopień naukowy, imię i nazwisko>

Katedra Informatyki Stosowanej

Instytut Nauk Technicznych

Toruń 2021

Pracę przyjmuję i akceptuję

.....

data i podpis opiekuna pracy

Potwierdzam złożenie pracy dyplomowej

.....

data i podpis pracownika dziekanatu

*UMK zastrzega sobie prawo własności niniejszej pracy magisterskiej (licencjackiej, inżynierskiej)
w celu udostępniania dla potrzeb działalności naukowo-badawczej lub dydaktycznej*

Spis treści

Spis treści	4
Summary/Abstract	5
Wstęp	6
Rozdział 1. Opis problemu	7
Co powinno znaleźć się w tym rozdziale?	7
Formatowanie tekstu	7
Wymagania.....	7
Krytyczny przegląd dostępnych rozwiązań.....	8
Podział zadań w przypadku pracy zespołowej z wyraźnie zaznaczonymi zadaniami autora pracy	8
Rozdział 2. Opis projektu	9
Co powinno być w rozdziale 2?	9
Błędy typograficzne	10
Cytowania.....	11
Sprawy formalne	11
Rozdział 3. Testy	12
Opis procedury testów	12
Wyniki testów.....	12
Testy użyteczności/ergonomiczności	12
Wnioski/Podsumowanie	12
Literatura.....	13
Dodatek A. Instrukcja obsługi programu.....	14

Automatycznie generowany spis treści na podstawie nazw rozdziałów i podrozdziałów.
Czcionka Times New Roman. Do pracy nie trzeba dołączać spisu rysunku i tabel.

Summary/Abstract

Obowiązkowe. Kilka zdań **w języku angielskim** opisujących projekt, jego cel i realizację. Ten opis będzie wykorzystany przy przesyłaniu pracy do APD (Archiwum Prac Dyplomowych).

Wstęp

Kilka zdań umieszczających projekt w **szerszym kontekście** rozwoju technologii informatycznych lub w kontekście społecznym. Jaki projekt może mieć wpływ na funkcjonowanie użytkownika? Czy przyspiesza procesy, wprowadza udogodnienia, czy też jest nowym podejściem do rozwiązania wcześniej istniejących problemów? Jakiego użytkownika może odnieść korzyści używając wytworzonych w ramach projektu rozwiązań? Jakiego **znaczenie** ma ten projekt dla jego użytkowników, zamawiającego i pozostałych tzw. interesariuszy projektu? A także dlaczego autor zdecydował się zająć tym projektem (**motywacja**)?

Wstęp nie powinien być zbyt długi (jedna, maksymalnie dwie strony) i nie powinien powielać wymienionych w pierwszym rozdziale technicznego opisu projektu, wymagań oraz przeglądu rynku (przeglądu konkurencyjnych rozwiązań). Może natomiast podkreślać wyjątkowość pomysłu i jego innowacyjność, będąc nietechnicznym wprowadzeniem do opisu projektu. Może też krótko podsumowywać zawartość całej pracy, przyjmując rolę streszczenia. Wówczas wcześniejsze *Summary/Abstract* może być tłumaczeniem tego fragmentu.

Rozdział 1.

Opis problemu

Co powinno znaleźć się w tym rozdziale?

Głównym tematem tego rozdziału jest cel pracy, czyli jasno sprecyzowane zadanie inżynierskie (ewentualnie badawcze), jakie ta praca ma rozwiązywać. Należy podkreślić, dlaczego zadanie nie jest trywialne – na czym polega jego trudność. Warto określić też potencjalne zastosowania i inną motywację do podjęcia się rozwiązania problemu, jeżeli nie została ona opisana we wstępie.

Formatowanie tekstu

Zalecamy użycie czcionki Times New Roman 12 lub 11. Tekst w akapitach powinien być wyjustowany (wyrównany z obu stron), interlinia równa 1,15. Odstęp przed akapitem równy 0 pkt, a po akapicie – 6 pkt. Do sformatowania nazwy rozdziału należy użyć stylu *Nagłówek 2*, a podrozdziałów – stylu *Nagłówek 3* (lub analogicznych stylów w LaTeX). Fragmenty kodu, słowa kluczowe języków programowania, nazwy zmiennych, klas, funkcji itp. – czcionka Courier New o rozmiarze 10 lub 9.

Wymagania

To obowiązkowy podrozdział, w którym opisane są wymagania stawiane projektowi.

Możliwy schemat opisu to:

- 1) Wprowadzenie zawierające m.in. definicje wszystkich używanych pojęć i akronimy
- 2) Ogólny opis produktu
 - a) kontekst funkcjonowania systemu (m.in. powiązania z innymi systemami),
 - b) charakterystyka użytkowników (ich role),
 - c) funkcje produktu (co mogą zrobić poszczególni użytkownicy),
 - d) ograniczenia (np. RODO).

Wszystko z jak największą liczbą diagramów.

- 3) Wymagania (np. według modelu FURPS):

- a) F – *functionality* – funkcjonalność

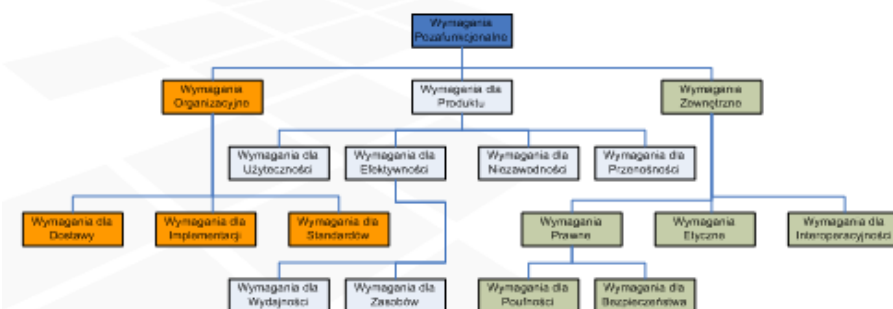
Określenie usług/funkcji, które powinno realizować i udostępniać oprogramowanie; w tym określenie sposobu, w jaki system będzie reagował na właściwe i niewłaściwe działania użytkowników (z różnymi rolami); tu także wymagania i preferencje dotyczące interfejsu użytkownika (UI). Warto omówić przypadki użycia (także diagramy).

- b) U – *usability* – użyteczność (rys. 1.1),
- c) R – *reliability* – niezawodność,

d) P – *performance* – wydajność,

e) S – *security* – bezpieczeństwo.

- Typy wymagań pozafunkcyjnych wg. Sommerville'a



Źródło: Sommerville, I. *Software Engineering*.
Eighth Edition. Addison-Wesley, 2004.

Rysunek 1.1. Wymagania niefunkcjonalne (źródło: <http://wazniak.mimuw.edu.pl/>)

Krytyczny przegląd dostępnych rozwiązań

Należy opisać dostępne rozwiązania, ich mocne i słabe strony i powód, przez który nie można ich użyć lub uzasadnienie, dlaczego mimo dostępności rozwiązania, autor podejmuje wysiłek stworzenia własnego. Tu powinny znaleźć się odwołania do dostępnych źródeł w postaci [1] lub [2-4], szczególnie w przypadku problemu badawczego.

Podział zadań w przypadku pracy zespołowej z wyraźnie zaznaczonymi zadaniami autora pracy

Ten podrozdział pojawi się tylko, jeżeli przedmiot pracy jest fragmentem większego projektu realizowanego przez kilku studentów. Zadania autora pracy muszą być jasno określone.

Rozdział 2.

Opis projektu

Każdy rozdział powinien składać się z krótkiego wprowadzenia (przed nazwą pierwszego podrozdziału)

Co powinno być w rozdziale 2?

W tym rozdziale należy szczegółowo opisać projekt i pracę wykonaną przez autora. To zasadniczy rozdział, który powie recenzentowi pracy, jak dużo pracy wykonał student i jak bardzo była trudna.

Za obowiązkowy należy uznać schemat modułów całego projektu (diagram klas z UML z zaznaczonymi modułami lub warstwami) oraz schematy wybranych zbiorów klas. W przypadku opracowania algorytmu, warto umieścić jego schemat blokowy. Ten rozdział może mieć formę dokumentacji technicznej, jeżeli tego zażąda promotor. Ewentualną dokumentację można również przenieść do osobnego rozdziału, a tu zostawić jedynie ogólny opis architektury systemu.

W przypadku opracowania lub implementacji algorytmu, należy tutaj umieścić jego opis razem z diagramem, oceną kosztu itp. Tu także należy opisać modele matematyczne używane w przygotowywanym projekcie.

W tym rozdziale należy umieścić szczegółowy opis wybranych fragmentów kodu z listingami i ewentualnie zrzutami ekranu. To dotyczy szczególnie tych fragmentów kodu, których opracowanie stanowiło trudność, a znalezione rozwiązanie może być interesujące dla kolejnych studentów. Ponumerowane listingi umieszczamy jako tekst z podpisem nad nimi (czcionka np. Courier, 10 lub 9). W listingach można wytłuszczyć słowa kluczowe, ale nie jest to obowiązkowe.

Listing 2.1. Przykładowy listing.

```
using System;

namespace PracaInzynierska
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello World!");
        }
    }
}
```

Również rysunki i tabele powinny być numerowane. Opis rysunków pod nimi, a tabel nad (podobnie jak listingów). Numeracje listingów, rysunków i tabel prowadzone są osobno. Nie trzeba dodawać ich spisów na końcu pracy. Odwołanie do rysunku: (rys. 2.1) lub (por. rys.

2.1). Do każdego rysunku, tabeli i listingu powinno być odwołanie w przylegającym do niego akapicie. Jeżeli rysunek nie został przygotowany przez autora pracy, należy podać źródło. W odwołaniach skrót „rys.” pisany małą literą (w odróżnieniu od angielskiego „Fig.”), z kropką na końcu. Nie należy używać określeń „poniższy lub powyższy rysunek/tabela”, odwołujemy się poprzez rysunki. W ostatecznej wersji pracy unikamy tzw. opisów-sierot (opis zostaje na końcu poprzedniej stronie, a rysunek lub tabela są już na następnej). Przed i po rysunku, listingu i tabeli warto zostawić jednoliniowy odstęp.

Tabela 2.1. Przykładowa tabela

L.p.	Nagłówek 1	Nagłówek 2
1	Treść 1	Treść 2
2	Treść 3	Treść 4

Uwaga! W tym i następnym rozdziale opisywana jest wyłącznie praca autora, nawet jeżeli projekt był realizowany przez zespół. W przypadku, gdy konieczne są informacje o elementach projektu realizowanych przez innych członków zespołu, należy wyraźnie zaznaczyć, że opis taki dotyczy kodu przygotowanego przez innego autora z podaniem referencji do jego pracy dyplomowej.

Szczególną uwagę należy poświęcić tym elementom projektu, których nie udało się zrealizować lub nie w pełni zrealizować. Warto usprawiedliwić czemu tak się stało.

Oto najbardziej popularne błędy składu, których należy unikać:

Wdowa – ostatni wiersz akapitu, który jest bardzo krótki (złożony z jednego krótkiego wyrazu).

Sierota – jedno- lub dwuliterowy spójnik pozostawiony na końcu wersu.

Szewc – pierwszy wiersz akapitu pozostawiony na końcu poprzedniej strony.

Bękart – ostatni wiersz akapitu, który został umieszczony na następnej stronie.

Część z tych błędów straciła na znaczeniu ze względu na powszechne użycie edytorów tekstu.

Należy także uważać, żeby podpisy rysunków, tabel i listingów znajdowały się na tej samej stronie co sam rysunek, tabela lub listing.

Zob. również listę błędów edycji, których należy unikać dostępną na stronie https://pl.wikipedia.org/wiki/B%C5%82%C4%99dy_typograficzne

Cytowania

Możliwe jest kilka sposobów podawania referencji do pracy cytowanych. W tej pracy użyto stylu z nawiasami kwadratowymi [1] lub [1-3]. Takie same oznaczenia powinny mieć prace, do których odwołuje się autor w części „Literatura” na końcu. Prace te powinny być ułożone w kolejności cytowania. W części „Literatura” nie powinno być prac, które nie są cytowane.

Inną metodą jest podawanie nazwiska i roku publikacji. Wówczas w części „Literatura” prace powinny być ułożone alfabetycznie według nazwisk. Przykłady:

Najważniejszym językiem programowania jest C++ (Stroustrup, 2014).

W książce Stroustrupa (2014) możemy znaleźć stwierdzenie, że C++ jest najważniejszym językiem programowania.

Uwaga! Fragment cytowany z innej pracy należy wyraźnie oznaczyć (cudzysłów lub kursywa) i podać źródło cytowania. W przeciwnym razie autorowi może być zarzucony plagiat. Prace przed zatwierdzeniem są weryfikowane przez system antyplagiatowy.

W pracy dyplomowej powinno znaleźć się przynajmniej kilkanaście rzeczywiście wykorzystanych odwołań do źródeł (cytowań), szczególnie w rozdziale 1, który opisuje problem. Problem badawczy powinien być dobrze osadzony w literaturze, a tezy, które nie są dowodzone w inny sposób - uzasadnione odwołaniami do literatury. Każde ogólne stwierdzenie powinno być rzetelnie uzasadnione przez autora bądź poparte podaniem referencji.

Sprawy formalne

Zgodnie z nową ustawą praca **nie musi być dostarczona w formie drukowanej**, lecz elektroniczną wersję pracy należy złożyć w serwisie Archiwum Prac Dyplomowych (<https://apd.umk.pl/>) wraz ze streszczeniem, słowami kluczowymi i tytułem w języku angielskim. Promotor może jednak poprosić o wersję drukowaną (we wniosku składanym przez egzaminem dyplomowym). Wówczas należy wydrukować wersję z numerem kontrolnym nadanym przez APD. Dopuszczalne jest dostarczenie drukowanej wersji pracy w miękkiej okładce, zbindowaną w dowolny sposób.

Cała praca powinna mieć 35 – 50 stron, włączając w to rysunki i listingi, ale to nie jest ścisły wymóg.

Rozdział 3.

Testy

Opis procedury testów

Jakie testy zostały przeprowadzone (poziom: modułowe, integracyjne, systemowe, akceptacyjne; typ: funkcjonalne, wydajności, bezpieczeństwa)

Wyniki testów

Tabele i wykresy z komentarzem

Testy użyteczności/ergonomiczności

Opcjonalne, na życzenie promotora

Wnioski/Podsumowanie

Wnioski szczegółowe i ogólne z przeprowadzonych testów. Nie trzeba pisać osobnego podsumowania jako rozdziału pracy.

Jeżeli projekt był przygotowany na potrzeby eksperymentu/symulacji, ten rozdział może mieć tytuł „Eksperyment” lub „Symulacja” i składać się z podrozdziałów: „Opis procedury eksperymentu”, „Hipotezy”, „Wyniki” i „Dyskusja”.

Literatura

- [1] J. Kowalski, *Tytuł książki*, Wydawnictwo, Miejsce wydania i rok wydania.
- [2] *Tytuł pracy*, Red. J. Kowalski (1-3 nazwiska), Wydawnictwo, Miejsce wydania i rok wydania. Praca zbiorcza
- [3] J. Kowalski, *Tytuł artykułu*, Tytuł czasopisma, Numer czasopisma, (Rok), Strony. Artykuł
- [4] <http://www.fizyka.umk.pl> – Strona Wydziału Fizyki, Astronomii i Informatyki stosowanej UMK w Toruniu. Strona internetowa
- [5] T. Penkala, *Podstawy chemii ogólnej*, PWN, Warszawa 1982.
- [6] C. Kittel, *Wstęp do fizyki ciała stałego*, PWN, Warszawa 1987.
- [7] G. Sęk, P. Sitarek, *Spektroskopia fotoodbiciowa*, Oficyna Wydawnicza Politechniki Rzeszowskiej, Rzeszów 1999.
- [8] J. Karwowski, W. Jaskólski, and J. Kobus, *Comment on "A comparison of relativistic and quasirelativistic line strengths" by A. K. Mohanty and D. H. Sampson*, *Physica Scripta* **38**, (1988), 554-556.
- [9] M. Bylicki, W. Jaskolski and R. Oszwaldowski, *Resonant tunnelling in asymmetrical double barrier structures - complex coordinate approach*, *Czech. Journal of Physics* **47**, (1997), 373.
- [10] A. Raczyński, J. Zaremba, *On spectra and population trapping in half-collisions*, *Physica Scripta* **44**, (1991) 258.

Dodatek A.

Instrukcja obsługi programu

Ten dodatek umieszczany jest w pracy tylko, jeżeli jest konieczny (tj. jeżeli obsługa programu jest szczególnie skomplikowana, a sam program może się przydać np. do zajęć lub następnym studentom piszącym prace dyplomowe) lub zażyczy sobie tego promotor.