



Politechnika Łódzka

Instytut Informatyki

INŻYNIERSKA PRACA DYPLOMOWA

MAPOWANIE OBIEKTOWO-RELACYJNE W JĘZYKU C++

Wydział: Fizyki Technicznej, Informatyki i Matematyki Stosowanej
Promotor: dr inż. Arkadiusz Tomczyk
Dyplomant: Marcin Maciaszczyk
Nr albumu: 165466
Kierunek: Informatyka
Specjalność: Inżynieria Oprogramowania i Analiza Danych

Łódź, 10 maja 2014r.

Instytut Informatyki

90-924 Łódź, ul. Wólczańska 215, **budynek B9**
tel. 042 631 27 97, 042 632 97 57, fax 042 630 34 14 email: office@ics.p.lodz.pl

Spis treści

1	Wstęp	5
1.1	Uzasadnienie wyboru tematu	5
1.2	Problematyka i zakres pracy	6
1.3	Cele pracy	8
1.4	Metoda badawcza	8
1.5	Przegląd literatury w dziedzinie	9
1.6	Układ pracy	9
2	Tytuł części teoretycznej	10
2.1	Podstawowe definicje	10
2.2	Istniejące rozwiązania w dziedzinie	10
2.2.1	Sprzęt	10
2.2.2	Oprogramowanie i wdrożone systemy	10
2.2.3	10
2.3	Wady i słabe punkty istniejących rozwiązań	11
2.3.1	Efektywność	11
2.3.2	Utrudniony dostęp	11
2.3.3	Wysokie koszty	11
3	Dalsze uwagi o edycji i formatowaniu pracy	12
3.1	Bibliografia i przypisy	12
3.2	Polskie akapity, cudzysłowy, itp.	13
3.3	Definicje i wyrażenia matematyczne	14
3.4	Jak wstawiać rysunki? tabele?	14
3.5	Listy wypunktowana i numerowana	15
3.6	Przenoszenie wyrazów	15
4	Technologie i metody użyte...	16
4.1	Sprzęt	16

4.1.1	Element 1	16
4.1.2	Element 2	16
4.2	Oprogramowanie	16
4.2.1	Serwer baz danych	17
4.2.2	Środowisko zintegrowane	17
4.2.3	Oprogramowanie klienckie	17
4.3	Technologie i metodologie programistyczne	17
4.3.1	Język programowania	17
4.3.2	Biblioteki	17
4.3.3	Wzorce projektowe	17
4.4	Inne, np. narzędzia i metody symulacji,	17
5	Biblioteka programistyczna Qubic	18
5.1	Analiza wymagań	18
5.1.1	Studium możliwości	18
5.1.2	Wymagania funkcjonalne	18
5.1.3	Ograniczenia projektu	18
5.2	Projekt	18
5.2.1	Projekt warstwy danych	18
5.2.2	Projekt warstwy logiki	19
5.2.3	Projekt warstwy interfejsu użytkownika	19
5.3	Implementacja: punkty kluczowe	19
5.4	Testy i wdrożenie	19
5.4.1	Testy wydajności	19
5.4.2	Testy regresyjne	19
5.4.3	Testy bezpieczeństwa	19
5.4.4	Dalsze testy	19
5.4.5	Testy...	19
5.5	Konserwacja i inżynieria wtórna	19
6	Podsumowanie	20
6.1	Dyskusja wyników	20
6.2	Ocena możliwości wdrożenia...	20
6.3	Perspektywy dalszych badań w dziedzinie	20
	Bibliografia	20
	Spis rysunków	21
	Spis tabel	22

<i>SPIS TREŚCI</i>	4
Załączniki	23

Rozdział 1

Wstęp

1.1 Uzasadnienie wyboru tematu

Wraz z rozwojem informatyki proces tworzenia oprogramowania wymaga od informatyków co raz większej wiedzy w poszczególnych dziedzinach takich jak na przykład bazy danych czy sieci komputerowe. Ciężko jest być specjalistą w każdej z tych dziedzin, dlatego też podczas tworzenia oprogramowania programiści co raz częściej sięgają po różnego rodzaju narzędzia programistyczne, które mają za zadanie ułatwić im pracę.

Przykładem takich narzędzi są aplikacje szkieletowe wykorzystywane jako fundamenty dla tworzonych aplikacji czy też biblioteki udostępniające zestawy określonych funkcji. Oczywiście nie zawsze mamy możliwość skorzystania z wspomnianych narzędzi, jednak jeśli taka istnieje warto wziąć to pod uwagę podczas fazy planowania tworzenia oprogramowania, ponieważ decydując się na korzystanie z nich jesteśmy w stanie zaoszczędzić sporo czasu, a także uniknąć wielu błędów związanych z niezajomością danej dziedziny.

Wraz z kolegą z tego samego roku studiów – Sebastianem Florkiem, zdecydowaliśmy się w ramach pisania pracy dyplomowej na wspólne stworzenie aplikacji szkieletowej, która będzie realizowała mapowanie obiektowo-relacyjne w języku C++. Wybór języka programowania związany był z dość dobrą jego znajomością, a także z pewnym doświadczeniem w programowaniu w tym języku nabytym w trakcie studiów. Mapowanie obiektowo-relacyjne jest to obecnie zagadnienie bardzo powszechne, szczególnie w językach programowania takich jak C# czy Java. Programiści C++ nie mają już tak dużego wyboru wśród dostępnych narzędzi służących do mapowania obiektowo-relacyjnego, co tłumaczy nasz wybór.

Podział naszej pracy zostanie opisany w dalszych rozdziałach, jednak na wstępie warto zaznaczyć, że tematem pracy Sebastiana jest generator opisu mapowania

obiektoowo-relacyjnego, który jest wstępnym etapem pracy naszej aplikacji szkieletowej. Celem wspomnianego generatora jest wygenerowanie pliku projektu, a także plików nagłówkowych oraz klas na podstawie istniejącej już bazy danych. Moim zadaniem będzie samo mapowanie obiektoowo-relacyjne, które będzie się opierało na wcześniej wygenerowanych plikach.

1.2 Problematyka i zakres pracy

Programowanie obiektowe jest obecnie jednym z najpopularniejszych paradygmatów programowania, a pojęcia takie jak klasa czy obiekt znane są wszystkim programistom. Podobnie jest z relacyjnym modelem organizacji baz danych i terminami takimi jak relacja czy krotka. Chcąc wykorzystać oba te podejścia w jednej aplikacji musimy zadbać o obustronną konwersję pomiędzy danymi z tabel relacyjnej bazy danych a obiektami aplikacji. Tym właśnie zajmuje się mapowanie obiektoowo-relacyjne, które wraz z tworzeniem aplikacji szkieletowych w języku C++ jest główną problematyką niniejszej pracy.

W tym momencie powstaje pytanie czy na prawdę warto korzystać z bibliotek i aplikacji szkieletowych służących do mapowania obiektoowo relacyjnego? Odpowiedź nie jest jednoznaczna w wszystkich przypadkach, ale warto wymienić jego podstawowe wady i zalety, których głębsza analiza znajduje się w dalszej części pracy. Zaczniemy od zalet wykorzystania narzędzi ORM¹:

- Nieobowiązkowa dla programisty znajomość języka SQL². W celu tworzenia zapytań do bazy danych korzystamy z interfejsu udostępnianego przez daną bibliotekę czy też aplikację szkieletową napisaną w wybranym przez nas języku programowania.
- Aby skorzystać z transakcji a także wielu innych funkcjonalności baz danych wystarczy zazwyczaj wywołać pojedynczą metodę.

Wszędzie gdzie pojawiają się zalety mamy też do czynienia z wadami, nie inaczej jest w tym przypadku:

- Konfiguracja narzędzi ORM jest najczęściej bardzo skomplikowana i wymaga sporo czasu aby była optymalna.

¹mapowanie obiektoowo-relacyjne (ang. Object-Relational Mapping)

²strukturalny język zapytań (ang. Structured Query Language)

- Proste zapytania są obsługiwane bardzo sprawnie, jednak gdy przetwarzamy duże ilości skomplikowanych zapytań wydajność nie dorówna nigdy zapytaniom napisanym przez specjalistę znającego doskonale język SQL.

Pros and cons

Using an ORM save a lot of time because :

DRY : You write your data model in only one place, it's easier to update, maintain and reuse the code. A lot of stuff is done automatically, from the database handling to I18N. It forces you to write MVC code, and in the end your app is cleaner.

Using an ORM is more flexible because :

It abstracts the DB system, so you can change it whenever you want. The model is weakly binded to the rest of the app, so you can change it or use it anywhere else. It let you use OOP goodness like data inheritance without head ache.

But ORM can be a pain :

You have to learn it, and they are not lightweight tools; It abstracts the DB. While it's ok if you know what's happening behind the scene, it's a trap for the noobs that can write very greedy statements, like a heavy hit in a for loop...

Głównym celem pracy jest analiza istniejących aplikacji szkieletowych oraz bibliotek programistycznych realizujących mapowanie obiektowo-relacyjne napisanych w języku C++ oraz zaproponowanie własnej aplikacji szkieletowej, zwanej także w dalszej części pracy Qubic. Zakres pracy to mapowanie obiektowo-relacyjne oraz tworzenie aplikacji szkieletowych w języku C++.

<http://stackoverflow.com/questions/1279613/what-is-an-orm-and-where-can-i-learn-more-about-it>

Niniejsza praca dotyczy zakresu inżynierii oprogramowania a także technologii baz danych. Głównym przedmiotem/celem pracy jest stworzenie/opracowanie/przeanalizowanie/zaprojektowanie.... **ten fragment pracy zawierać musi wyraźne określenie problemu badawczego oraz jego pogłębioną analizę**

Dlaczego podejmowanie tej tematyki jest potrzebne? Czy są inne rozwiązania tego problemu/tych problemów? Jakie? czy są lepsze/gorsze, tańsze/droższe, itp.

Przed jakimi wyzwaniami stoi osoba podejmująca tematykę?

Określić spodziewane efekty pracy: W wyniku doświadczeń przeprowadzonych w zakresie pracy polepszeniu uległo.... podać konkretne wskaźniki rezultaty, jak np. przyspieszenie obliczeń, redukcja kosztu, nowe oprogramowanie itp. w złotych, sekundach, procentach, roboczogodzinach itp.

(razem max. 3 strony - strona przeliczeniowa = 1800 znaków, średnio 30 wierszy po 60 znaków)

1.3 Cele pracy

Wymienić w punktach cele pracy, rozpoczynając od celów poznawczych (dotyczących zebrania wiadomości, przybliżenia/popularyzacji technik / metod / zagadnień). W drugiej kolejności wyienić cele praktyczne.

Przybliżenie (popularyzacja) metody/technologii/systemu...

Propozycja rozwiązania problemu....

Opis zastosowania technologii X w problemie Y...

Przedstawienie prototypu systemu/układu/aplikacji...

Określenie przydatności algorytmu Z do rozwiązania problemu T.....

Opracowanie strategii ... w celu poprawy wydajności/jakości...

Ocena możliwości wdrożenia proponowanych rozwiązań, ich wartość praktyczna, lokalne i globalne możliwości zastosowania

itp.

Każdy cel opisać w minimum 2-3 zdaniach. Użyte określenia muszą być powszechnie zrozumiałe, nie stosujemy skrótów, slangu, tzw. makaronizmów, np. „softler”.
cały podrozdział ok. 1 strony przeliczeniowej czyli 1800 znaków

1.4 Metoda badawcza

- Studia literaturowe
- Analiza budowy i działania istniejących produktów
- Projektowanie i prototypowanie nowatorskich rozwiązań
- Obliczenia i

Każdy element opisać w minimum 2-3 zdaniach. Np. studia literaturowe powinny odnosić się do charakterystyki wykorzystanych źródeł książkowych, czyli: Jaka jest podstawowa literatura dziedziny, czy jest dostępna w języku polskim, czy trzeba je tłumaczyć, czy wiedza na ten temat jest zebrana w jednym miejscu, czy jej synteza jest osobnym zadaniem itp. Jak duży jest udział źródeł elektronicznych w tej „dziś” wiedzy i badań, itd.

Jakie metody badawcze są typowe dla danego tematu. Dlaczego je zastosowano, ewentualnie dlaczego zastosowano inne?

WYMAGANE ODNOŚNIKI DO POZYCJI BIBLIOGRAFII.

cały podrozdział ok. 1 strony przeliczeniowej czyli 1800 znaków.

1.5 Przegląd literatury w dziedzinie

Rozszerzyć odpowiedni podpunkt z metody badawczej, np. wg podziału:

Źródła książkowe polskojęzyczne i tłumaczenia

Źródła książkowe obcojęzyczne

Artykuły naukowe, raporty z badań, komunikaty konferencyjne, dokumentacje techniczne, manuale, instrukcje

Źródła elektroniczne

1.6 Układ pracy

Tematem pracy jest:, zaś za główny cel przyjęto

Rozdział zawiera wstęp i cele pracy. W rozdziale drugim opisano/..... w Rozdziale 3. zawarto..... Rozdział 4. przedstawia.....

W podsumowaniu pracy przedstawiono....., z czego wynika, że

Najważniejszym wnioskiem/wynikiem/rezultatem pracy jest.....

wyraźnie określić CO TO JEST.

cały podrozdział ok. 1 strony.

Rozdział 2

Tytuł części teoretycznej

2.1 Podstawowe definicje

Ten podrozdział powinien zawierać dokładny opis terminologii pojęć zasadniczych dla tematu pracy, którymi autor będzie się posługiwał przy realizacji głównych celów pracy.

2.2 Istniejące rozwiązania w dziedzinie

W tym podrozdziale zostaną opisane.....

2.2.1 Sprzęt

.....

2.2.2 Oprogramowanie i wdrożone systemy

.....

2.2.3

.....

2.3 Wady i słabe punkty istniejących rozwiązań

2.3.1 Efektywność

.....

2.3.2 Utrudniony dostęp

.....

2.3.3 Wysokie koszty

.....

Rozdział 3

Dalsze uwagi o edycji i formatowaniu pracy

Pracę w \LaTeX ’u najlepiej składać w szablonie `report`, ze względu na jendostronny wydruk (jak w `article`) i możliwość dzielenia pracy na rozdziały, a co za tym idzie, tworzenia spisu treści, spisu tabel, rysunków.

Przykład 3.1 *Przykład*

Wniosek 3.1 *Wniosek*

3.1 Bibliografia i przypisy

Spis literatury dołącza się w \LaTeX ’u automatycznie na końcu pracy (zob. komenda `beginthebibliography`). Informacje o sposobie cytowania zawarte są na stronie Biblioteki Głównej PŁ także udostępnione na <http://ics.p.lodz.pl/~aniewiadomski>.

Przykład cytowania – jak podaje praca [1],, jednakże autorzy [2] twierdzą, iż.....

Za każdym razem, kiedy w pracy pojawia się treść na podstawie jakiegoś tekstu źródłowego czyjegoś autorstwa, oznaczamy takie miejsce przypisem¹. Przypis zawierać musi numer jakim w spisie literatury, czyli bibliografii, oznaczono tę pracę, np. tak². **Wszystkie źródła tekstów, rysunków, danych, wykresów, schematów,**

¹Treść przypisu pierwszego

²[3], ss. 3–6 (czyli praca trzecia w spisie literatury, wykorzystany fragment znajduje się na stronach od 3. do 6.)

kodów i informacji wykorzystanych w pracy muszą być zamieszczone w bibliografii. Wszystkie pozycje literatury zamieszczone w bibliografii muszą być cytowane w treści pracy, na dowód, iż zostały rzeczywiście użyte przy pisaniu pracy.

Źródła elektroniczne

Źródła elektroniczne, zwłaszcza internetowe należy cytować z należytą uwagą na ich jakość. Nie cytujemy źródeł wątpliwej jakości lub wtórnie przekazujących czy też powielających wiedzę zawartą w innych źródłach, np. fora internetowe lub wikipedia.

Wszystkie wykorzystane źródła elektroniczne powinny być przez Autora pracy skopiowane w dniu ich wykorzystania i dołączone np. na CD/DVD do wersji drukowanej pracy.

Odnośniki do źródeł elektronicznych muszą zawierać pełną ścieżkę, np. do pliku lub rysunku, a nie jedynie domenowy adres portalu, np.

http://serwer.com/temat/podtemat/katalog/plik_strony.html (stan na dzień: 2009-12-05)

ale nie

www.portal.pl. (!!!!!)

Niedochowanie tego wymogu może stać się powodem odrzucenia pracy ze względów formalnych („brak możliwości weryfikacji źródeł wykorzystanych w pracy”).

3.2 Polskie akapity, cudzysłowy, itp.

Akapity stosujemy zawsze z wcięciem, ale bez wiersza odstępu pomiędzy akapitami. Ta forma jest przyjęta dla publikacji polskojęzycznych. **W szczególnych przypadkach (także w tym szablonie)** akapit występujący bezpośrednio po tytule rozdziału, sekcji, podsekcji itp. NIE JEST WCIĘTY.

Ten akapit JEST WCIĘTY. NIE MA także PUSTEGO WIERSZA pomiędzy tym akapitem a poprzednim.

Podobne uwagi dotyczą wszystkich innych elementów formatowania pracy – muszą być zgodne ze zwyczajami przyjętymi W JĘZYKU POLSKIM. Np. cudzysłowy wyglądają tak: „cudzysłów”, ale nie ”cudzysłów”, albo też ‘cudzysłów’ czy “cudzysłów”.

Rys. 3.1: Funkcja przynależności zbioru rozmytego – Podpis ZAWSZE POD rysunkiem, numeracja w postaci #.##.

(wypada podać źródło, czyli literaturę, z której rysunek pochodzi, ewentualnie *opracowanie własne.*)

3.3 Definicje i wyrażenia matematyczne

Definicja 1 *Niech X będzie przestrzenią....*

Do definicji odnieść się można poprzez jej etykietę: jak podano w Def. 1

Przykładowe podkreślenie... tekst podkreślony, pogrubienie: **tekst pogrubiony** oraz wyróżnienie *tekst wyróżniony, czyli kursywa*. Dalszy tekst rozdziału Dalszy tekst rozdziału Dalszy tekst rozdziału Dalszy tekst rozdziału Dalszy tekst rozdziału Dalszy tekst rozdziału Dalszy tekst rozdziału a teraz koniec linii...

... i nowy akapit. Akapity muszą być standardowo wcięte.

Przykład wzoru matematycznego numerowanego

$$E = m \cdot c^2 \quad (3.1)$$

Wszystkie symbole matematyczne występujące w tekście „na bieżąco”, czyli nieoznaczone numerem równania TAKŻE PISZEMY W TRYBIE MATEMATYCZNYM, CZYLI K U R S Y W A : $a = b \cdot c$, ale nie: $a = b * c$ (!!)

Numeracja wzoru – ZAWSZE w POSTACI (#.##) Jak podaje wzor (3.1).... (koniec linii).

Wyrażenia matematyczne można też wpisywać w wierszu – używamy wówczas znaku '\$', który rozpoczyna i kończy wyrażenie, np. wg Einsteina $E = m \cdot c^2$...

3.4 Jak wstawiać rysunki? tabele?

A teraz pora na rysunek:

Rysunki i tabele nie powinny przekraczać 0.9 szerokości tekstu i zasadniczo powinny występować na górze strony.

Odnosić się do rysunku można poprzez jego etykietę "label", np. jak widać na rys. 3.1.....

Tab. 3.1: Tytuł tabeli ZAWSZE NAD TABELĄ, numeracja w formie #.##. (wypada podać źródło, czyli literaturę, z której tabela pochodzi, ewentualnie *opracowanie własne*.)

Alg.	tytuł kolumny 1	tytuł kolumny 1	Tytuł kolumny 3
a	b	c	d	e
a	b	c	d	e

Jak widać, rysunek nie wypada w dokumencie w tym samym miejscu co w kodzie, choć czasem się tak zdarza. Jeśli potrzebujesz przenieść rysunek, zajrzyj do rozdziału 2.11. manuala pt. *Wstawki*.

3.5 Listy wypunktowana i numerowana

- pierwszy element listy wypunktowanej
- drugi...
- trzeci...

Nowy akapit z listą numerowaną.

1. pierwszy element listy NUMEROWANEJ
2. drugi...
3. trzeci...
4. trzeci...
5. trzeci...

3.6 Przenoszenie wyrazów

Skorzystaj z polecenia `hyphenation`

w preambule dokumentu, lub dziel wyrazy „ręcznie” czyli właśnie tak jak tu: podzielone wyrazy.

Rozdział 4

Technologie i metody użyte w części badawczej

Tytuł tego rozdziału ma dwie wersje: zwykłą, (w kodzie: w nawiasach klamrowych), która pokazuje się na stronie rozpoczynającej rozdział, oraz krótką (w kodzie: w nawiasach kwadratowych), która pokazuje się w spisie treści i w nagłówku

W rozdziale 2 podano podstawy teoretyczne i ogólny zakres pracy. W niniejszym rozdziale opisana zostanie technologia XYZ oraz metoda ABC użyta w części praktycznej, patrz rozdział 5.

4.1 Sprzęt

.....

4.1.1 Element 1

.....

4.1.2 Element 2

.....

4.2 Oprogramowanie

.....

4.2.1 Serwer baz danych

.....

4.2.2 Środowisko zintegrowane

.....

4.2.3 Oprogramowanie klienckie

4.3 Technologie i metodologie programistyczne

.....

4.3.1 Język programowania

.....

4.3.2 Biblioteki

.....

4.3.3 Wzorce projektowe

.....

4.4 Inne, np. narzędzia i metody symulacji,

Rozdział 5

Biblioteka programistyczna Qubic

Ta część pracy może być podzielona na więcej rozdziałów, np. kiedy autor chce w szczególności podkreślić któryś z etapów projektu. W zależności od tematu i celów pracy, pewne sekcje można dodać (np. przy projektowaniu sieci, instalacji i konfiguracji serwerów usług sieciowych), inne zaś pominąć.

5.1 Analiza wymagań

5.1.1 Studium możliwości

5.1.2 Wymagania funkcjonalne

.....

5.1.3 Ograniczenia projektu

5.2 Projekt

5.2.1 Projekt warstwy danych

1. normalizacje baz danych
2. projekt bazy/baz
3. grupy użytkowników i ich prawa dostępu do danych (zależne od implementacji bazy)
4. ew. diagramy klas warstwy danych

5.2.2 Projekt warstwy logiki

1. Diagramy i scenariusze przypadków użycia
2. Diagramy przepływu danych (lub ich odpowiedniki)
3. ew. diagramy klas, wzorce projektowe itp.

5.2.3 Projekt warstwy interfejsu użytkownika

Wybór środowiska i platformy działania

Rodzaj aplikacji (klient-serwer, thick/thin client, aplikacja „biurkowa”, usługa, klient hybrydowy, itp.

Technologie projektowania i realizacji interfejsu użytkownika, np. biblioteki

5.3 Implementacja: punkty kluczowe

5.4 Testy i wdrożenie

5.4.1 Testy wydajności

5.4.2 Testy regresyjne

5.4.3 Testy bezpieczeństwa

5.4.4 Dalsze testy

5.4.5 Testy...

5.5 Konserwacja i inżynieria wtórna

Jak przebiega eksploatacja systemu/projektu? Jakie wady i zalety ujawniły się po np. 2-miesięcznym okresie testowania i użytkowania?

Jak można skorzystać z tej wiedzy praktycznej pod kątem rozbudowy pracy? Jakie elementy systemu powinny zostać w pierwszej kolejności zmodyfikowane?

Rozdział 6

Podsumowanie

6.1 Dyskusja wyników

Dzięki zrealizowaniu pracy poprawie uległa wydajność Ponadto, o ?? % skrócony został czas, a koszty osiągnięcia zamierzonego efektu zostały zmniejszone z ???pln do ???pln za godzinę/ dzień/ jednostkę sprzętu.....

Które cele pracy udało się zrealizować? co z tego wynika? Które cele pracy pozostały niezrealizowane i dlaczego?

6.2 Ocena możliwości wdrożenia proponowanych rozwiązań...

... ich wartość praktyczna, lokalne i globalne możliwości zastosowania, kwestia praw autorskich do powstałych produktów, itp.

6.3 Perspektywy dalszych badań w dziedzinie

Jak można kontynuować tę pracę, zwłaszcza pod kątem studiów uzupełniających magisterskich i/lub doktoranckich. Co jeszcze powinno być zrobione lub ulepszone? Co należy zmienić lub poprawić w pracy z dzisiejszego punktu widzenia?

Bibliografia

- [1] Kacprzyk J. (1986) Fuzzy sets in system analysis. PWN, Warsaw (in Polish).
- [2] Kacprzyk J., Strykowski P. (1999) Linguistic Data Summaries for Intelligent Decision Support, Proceedings of EFDAN'99. 4-th European Workshop on Fuzzy Decision Analysis and Recognition Technology for Management, Planning and Optimization, Dortmund, 1999, 3–12.
- [3] Kacprzyk J., Yager R. R. (2001) Linguistic summaries of data using fuzzy logic. *International Journal of General Systems* 30:133–154

Spis rysunków

3.1	Funkcja przynależności zbioru rozmytego – Podpis ZAWSZE POD rysunkiem, numeracja w postaci #.##.	14
-----	---	----

Spis tabel

- 3.1 Tytuł tabeli ZAWSZE NAD TABELĄ, numeracja w formie #.##.
(wypada podać źródło, czyli literaturę, z której tabela pochodzi,
ewentualnie *opracowanie własne.*) 15

Załączniki

1. Załącznik nr 1
2. Załącznik nr 2
3. Załącznik nr 3